# Enthusiast'99®

OFFICIAL PUBLICATION OF THE INTERNATIONAL 99/4 USERS GROUP

International 99/4 Users Group

JANUARY 1984
Vol. 2, Number 1


## A WOMAN'S VIEW— QUILT SQUARES

## STRUCTURED PROGRAMMING PAYS OFF

# SCOTT, FORESMAN PROBE

# EASY AS PIE! IT'S BASIC

## Unlock the full potential of your computer, and satisfy your family's learning appetite.

PROBE makes beginning Basic Programming as easy as pie.

Scott, Foresman's approach blends hands-on training with self-instructional activities. From the first minute, PROBE's easy to follow format shows you how to turn your computer into a productive working tool. Learn simple exercises like "Teaching the Computer to COUNT." Discover how to DEBUG. Create exciting graphics, sounds, and animation.

There's even a Helper's Manual with answers to all the questions.

PROBE is available for four different age levels, ranging from five years to adult. There's nothing else on the market like PROBE.

**SCOTT, FORESMAN . . .**
***Products with tomorrow in mind.***


**THERE'S MORE TO PROBE THAN MEETS THE EYE.**

Workbook, Helper's Manual, a wall chart of computer commands, 22" x 16" keyboard wall chart, 50-sheet pad of screen grids, and diskette. Available for six popular personal computers.

Buy PROBE wherever quality software is sold or write:

**Scott, Foresman and Company**

**Electronic Publishing Division**

1900 East Lake Avenue Glenview, Illinois 60025

# DHEIN'S

## CONTINUES TO SUPPORT TI-99/4A OWNERS

## NOW IN STOCK

### PERIPHERAL EXPANSION BOX
### AND ALL PERIPHERALS

(Check for availability as they are going fast)

### MOST TI CARTRIDGES, TAPES AND DISKS STILL AVAILABLE

## LOTS OF THIRD PARTY PRODUCTS NOW IN STOCK — MUCH MORE TO COME!

SEND COUPON OR POSTCARD FOR **FREE** COMPLETE DESCRIPTIVE LISTING

COUPON—MAIL TODAY

**FREE** COMPUTER BULLETIN

Name _____

Street _____ State _____

City _____ Zip _____

**The best combination of Price, Service and Quality is TRUE VALUE.**
**More than just a name, it is our way of doing business.**

# DHEIN'S

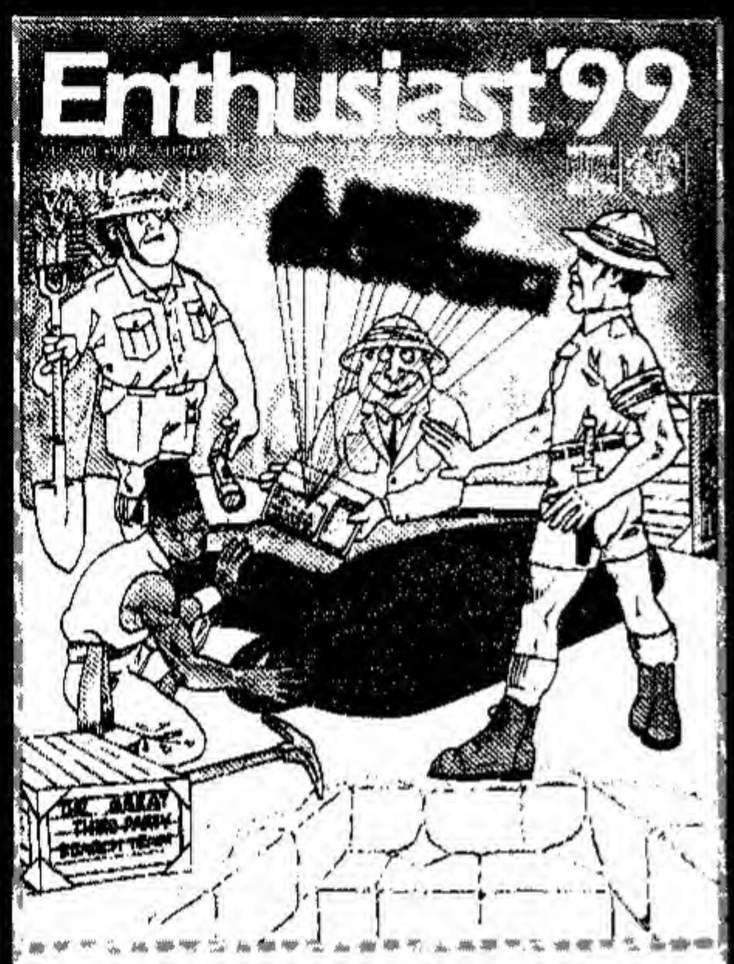
True Value HARDWARE STORE

(319) 236-3861

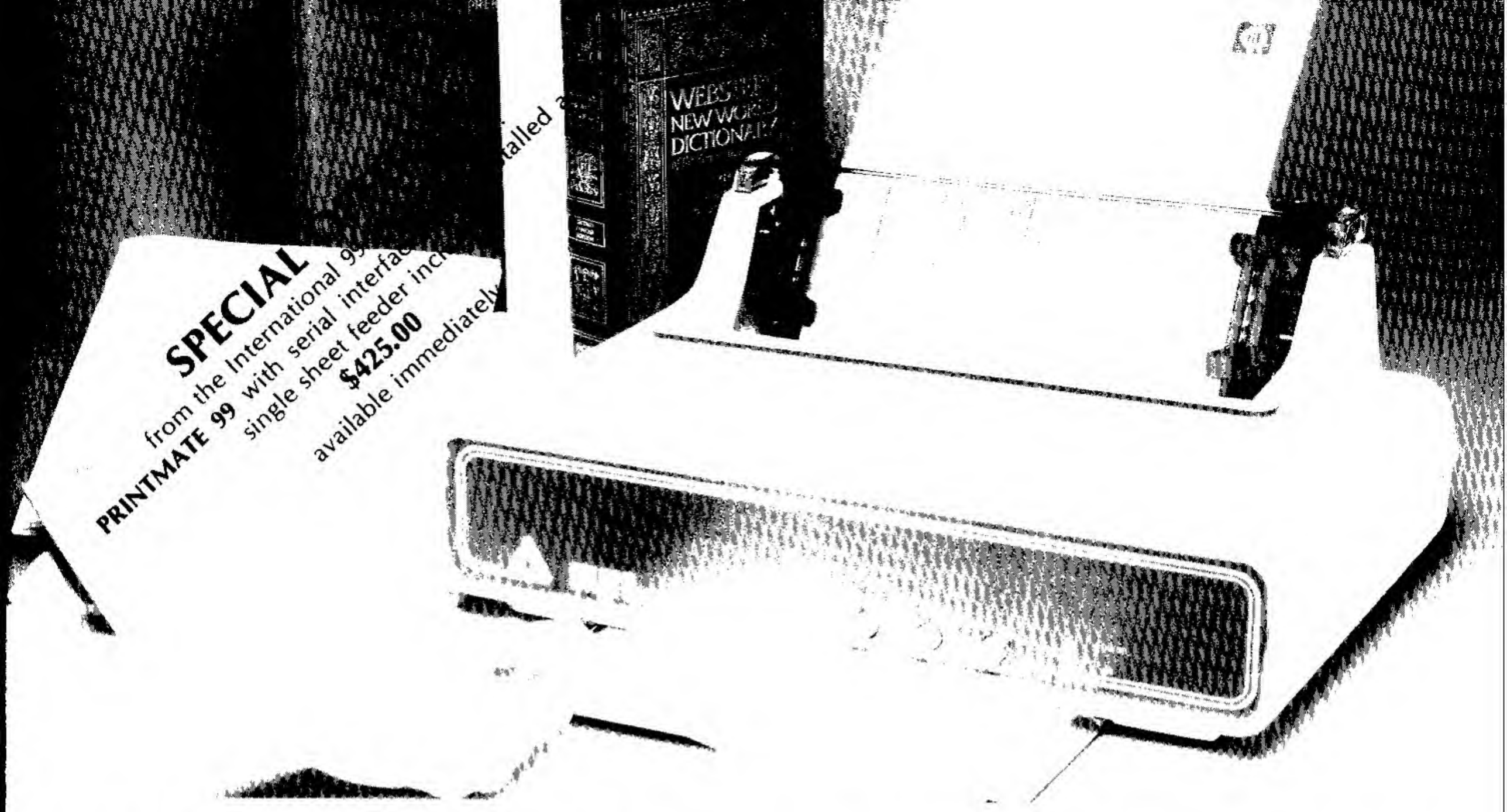
7 W. Airline Hwy.
Waterloo, IA 50701

# Enthusiast'99™

January 1984
Vol. 2, Number 1




**Enthusiast '99** magazine (ISSN 0737-9013) is published bi-monthly by LaPUBCO and distributed by the International 99/4 Users-Group, Inc., P.O. Box 67, Bethany, OK 73008. Editorial office is located at 5400 N.W. 23rd St., Suite 200, Oklahoma City, OK 73127, tel.(405) 948-1023. Office hours are Monday through Friday, 8:30-5:30, central time. **Enthusiast '99** is published for members of the International 99/4 Users-Group as part of their membership dues; annual subscriptions not available. Individual copies may be acquired for $2.75 each, U.S. funds only. Send address changes to P.O. Box 67, Bethany, OK 73008. Members wishing to send editorial correspondence to **Enthusiast '99** should address same to **Enthusiast '99**, P.O. Box 67, Bethany, OK 73008. Unaccepted manuscripts will not be returned unless accompanied by sufficient first class postage and self-addressed envelope. **Enthusiast '99** cannot be responsible for lost manuscripts, photographs, or program media. Opinions expressed by the authors are not necessarily those of **Enthusiast '99. Enthusiast '99** reserves the right to refuse advertising. Each separate contribution to this issue and the issue as a collective work copyright © by the International 99/4 Users-Group, all rights reserved. No claim is made by either **Enthusiast'99** or the International Users-Group to exclusive use of 99 apart from the mark as shown. Where necessary, permission is granted by the copyright owner for libraries and others registered with the Copyright Clearance Center (CCC) to photocopy any article herein with a base fee of $1 plus .25 cents per page per article or item. Payments should be sent directly to CCC, 21 Congress Street, Salem, MA 01970. Copies made for other than personal or internal reference use without permission of the International 99/4 Users-Group is prohibited by law; contact **Enthusiast '99** for permission and instruction. Requests for special permission or bulk orders should be addressed to the publisher. 99/4 Home Computer Users-Group, International 99/4 Users-Group and **Enthusiast '99** are registered trademarks of the International 99/4 Users-Group.




# PrintMate™ 99

## THE MOST ADVANCED PRINTER IN ITS CLASS

Today's microcomputer and small business applications are more demanding and complex than ever before, requiring a new breed of printer that can call on capabilities designed to enhance your expression on paper. The PrintMate™ 99 is designed with the features to solve your print needs while increasing your computer's thruput. In fact, this printer mates so well with your applications that you'll agree that it should be called a "PrintMate™."

**HIGH SYSTEM THRUPUT** — PrintMate™ 99 speeds through each task at 100 characters per second. All printing, interface and paper handling functions are under direct control of PrintMate™ 99's microprocessor array. A quick cancel feature provides thruput of up to 150 lines per minute on short lines and the unprinted line slew rate is 600 lines per minute.

**EXPANDABLE PRINT BUFFER** — PrintMate™ 99 has a standard 1K buffer memory and is optionally expandable to 2K. In either configuration, PrintMate™ 99 will print while simultaneously accepting data from your computer, minimizing your micro's waiting time.

**STANDARD POPULAR INTERFACES** — PrintMate™ 99 is plug compatible with most popular microcomputers. Each PrintMate™ 99 is equipped for both parallel (Centronics) and serial (RS232C) interface. Other special interfaces are optionally available.

**CORRESPONDENCE QUALITY** —PrintMate™ 99 offers near letter quality printing with its 11x9 "serif-style" dot matrix correspondence font. The 7x9 dot matrix allows high speed printing at either 10, 12 or 17 characters per inch enabling up to 136 columns to be printed on standard letter width paper. Double wide characters are software selectable and can be intermixed on a line for message highlighting. Crisp printing is assured with a long life cartridge ribbon which can be changed in seconds.

**PAPER AND FORMS VERSATILITY** — Printmate™ 99 offers exceptional versatility in handling the paper, forms or special media that best suit your applications. Both friction and tractor feed are yours at the flick of a lever. Paper may be from 1 to 9.5 inches in width and multiple copy forms may be as thick as .012 inch without printhead adjustment. Paper and forms can be fed from the front, bottom or rear of PrintMate™ 99 to suit the convenience of your workstation.

**SINGLE SHEET INSERTION** — PrintMate™ 99's single sheet feeder option lets you insert single sheets of paper from the front of PrintMate™ 99 more easily than you load a typewriter. With the single sheet feeder and correspondence font, PrintMate™ 99 makes letter writing and printing on cut forms such as invoices simpler than ever.

**TRACTOR FEED** — The adjustable tractors give precise control when using fan-fold paper, pre-printed forms and continuous labels. PrintMate™ 99's straight paper path performs trouble free printing on self-adhesive labels.

**FRICTION FEED** — PrintMate™ 99's friction feed allows the use of paper and other media that is not punched for tractor feeding. Roll paper may be easily used with the PrintMate™ 99.

**GRAPHICS** — High resolution, dot addressable graphics capability is included as a standard feature for applications using plotting, printing of screen graphics, special fonts and characters such as logos. The standard graphics capabilities of the PrintMate™ 99 allow the printing of 6120 individually addressable dots per square inch offering excellent resolution of graphics and special characters. Horizontal dot spacing is easily set to one of four dot densities.

**SPECIALIZED GRAPHICS PACKAGES** — PrintMate™ AP-PAK™ applications packages are designed to enable most popular micro-computers to excel in using PrintMate™ 99's special graphics and character capabilities. With PrintMate™ 99 and an AP-PAK™ you can create and edit your own special characters or utilize any one of several fonts. Graphs and bar charts are easily defined and printed using friendly AP-PAK™ software.

**SOFTWARE AND HARDWARE CONTROL** — Most PrintMate™ 99 functions may be controlled either by printer switch settings or by simple commands direct from your micro-computer. This enables you to print a variety of typestyles and sizes, graphs and special characters continuously under full program control. The PrintMate™ 99 is the responsive performer that perfectly mates with your micro-system or personal computer.

The Printer People
mpi™

**International 99/4 User's—Group**
P.O. Box 67
Bethany, Oklahoma 73008
Phone 1-405-948-1023

# NEWS BYTES

### COMMODORE ANNOUNCES NEW COMPUTERS

While some personal computer manufacturers are seen phasing out their computer line, Commodore plans to continue with the $200 Commodore 64 and under-$100 VIC-20. Sales are apparently so good for Commodore that the firm introduced two new computers, the 264 and 364 lines, at the January Consumer Electronics Show.

Claiming an impressive $1 billion-plus in sales for calendar year 1983, Commodore announced that its new computer lines will begin shipping in April. While no price has been quoted for either product, retailers expect each of them to boast a $399 price tag. Neither computer is software-compatible with the older units.

The 264 line will join the already-available SX-64 portable as a member of Commodore's new family of computers.

Additionally, Commodore announced several new plans during the CES: talking software and a new speech module, agreements with several third party software firms such as Digital Research and MicroPro, a venture into the software/books market, and the debut this spring of a computer with touch screen capability.

### SIERRA ON-LINE, WALT DISNEY REACH SOFTWARE AGREEMENT WITH TI

Under a joint agreement announced at the January Consumer Electronics Show with Texas Instruments and Walt Disney, Sierra On-Line will become the first third-party software publisher to manufacture and market software for the 99/4A Home Computer.

Sierra On-Line will supply five educational software programs for the 99/4A which feature such Disney characters as Peter Pan, Pinnochio, Winnie the Pooh and Professor Von Drake. The programs were developed by Disney for TI under an earlier production and marketing deal. Disney will distribute the programs to the school market.

The programs will also be made available for IBM, Apple, Commodore and Coleco's Adam.

### DATABAR CONTINUING OSCAR, DEVELOPING SOFTWARE SERIES

In the wake of Texas Instruments' decision to discontinue the production of the TI 99/4A Home Computer and associated software, Databar Corporation has announced that it is developing an extensive network of software programs to support 99/4A owners.

"There is an installed base of over two million systems, and those users need software," said Databar President Les Arnold. "Databar is fully committed to meeting their on-going need for programs by providing software in eight categories specifically for the 99/4A user."

The programs utilize OSCAR, a bar-code scanning device priced at around $65, and available from Databar Corporation or through the International 99/4 Users-Group. Each program from Databar's software series is approximately four pages long; data entry time, however, is cut in half utilizing OSCAR. The series covers the following subject areas: science and math, home legal, health, games, word skills, home business, educational and BASIC program writing skills.

Databar software is published in bar code format and retails for $9.95 per package and by subscription to Databar, 10202 Crosstown Circle, Eden Prairie, MN 55344, (612) 944-5700.

### COMMODORE FOUNDER RESIGNS

Jack Tramiel, founder of Commodore International Ltd, shocked the Home Computer industry Friday, January 13 when he resigned his post as president and chief executive officer of the home computer company.

**Stating that he believes Commodore needs a "professional executive" to take it beyond the $1 billion annual sales mark, Tramiel stepped down from one of the few home computer companies able to boast a profit in sales for 1983.**

Commodore is currently negotiating with Marshall F. Smith, 55, CEO of the U.S. unit of the Netherlands-based Thyssen-Bornemisza, to become president and CEO of the Home Computer company, which in three months ending December 31 made and shipped more than two million products including software programs, home computers and accessories.

Based in New York since 1974, Thyssen-Bornemisza has interests in shipbuilding, transport containers and metal and automotive parts. While the firm has little experience in computer-market expertise, Smith has the manufacturing experience Commodore says it seeks. Commodore said it has chosen a successor whom it will announce at the end of January.

### MATTEL SELLS MARKETING RIGHTS OF AQUARIUS

Mattel, Inc. has sold the marketing rights to its Aquarius home computer to the system's producer, the Radofin Electronics (Far East) Ltd. unit of London-based Fobel International.

No terms have been disclosed by either Mattel or Radofin, which assembles the Aquarius machines in Hong Kong. A spokesman said Mattel plans to continue supplying software for the Aquarius system.

Previously, Mattel had said it would retain domestic and United Kingdom marketing for Aquarius while pursuing sale of rights in non-English speaking international markets. The transaction with Radofin, however, involves all marketing rights, the Mattel spokesman said.

Last September, Mattel cited severe competition in the electronics business to blame for the firm's large losses at that time. It said it would sharply decrease marketing activity for the Aquarius and reduce the size of the entire operation, but continue to support the Intellivision software and hardware businesses.

For the nine months ended October 29, Mattel reported a $222.8 million net loss and said its net worth had shrunk to $26.7 million from $291.4 million a year earlier. A Radofin company spokesman confirmed that it will market current inventory displaying a Mattel nameplate, then distribute subsequent systems bearing no company name.

### IMAGIC TO MARKET FIVE 99/4A PROGRAMS

Texas Instruments and Imagic have announced that Imagic will assume marketing responsibilities for five software packages developed for the TI 99/4A Home Computer. Originally, TI had obtained manufacturing and marketing rights to Demon Attack, Microsurgeon, Fathom, Wing War and Moonsweeper.

"We are pleased that Imagic has agreed to market these five Imagic software games previously announced by TI for the 99/4A," said Dale Osborn, Manager in charge of phaseout of TI's Home Computer operations. "This agreement is yet another step by TI to provide TI 99/4A Home Computer owners with additional sources of software for their computers."

### TI INTRODUCES NU MACHINE

Texas Instruments has introduced Nu Machine™, the first commercial computer system implementing the advanced, high-speed 32-bit NuBus™ technology developed at the Massachusetts Institute of Technology.

Nu Machine may be configured with a 68010-based processor with cache memory and/or user-designed processors, UNIX™-based operating system, high resolution graphics display, high-performance mass storage peripherals, and a MultiBus™ subsystem.

Nu Machine's capabilities make it well-suited to applications such as VLSI circuit design, computer-aided engineering, printed circuit board design, mechanical design, simulation and advanced software development.

It is offered in two configurations; a small office unit designed to operate in a quiet environment and a rackmount model that supports larger peripherals and is designed for computer room operations. The office model is configured with a 68010 CPU, 512 kilobyte memory, 84 megabyte disk, ¼-inch cartridge tape, display, keyboard and mouse and is priced at $36,240 at OEM quantity 25. A similarly configured rackmount system with a 474 megabyte disk and ½-inch streamer tape is priced at $53,470 at OEM quantity 25.

(Nu Machine and NuBus are trademarks of Texas Instruments Incorporated; MultiBus is a trademark of Intel Corporation; UNIX is a trademark of Bell Laboratories.)

### ADAM SEES PRODUCTION PROBLEMS, COLECO SEES FOURTH QUARTER LOSS

Production delays with Coleco's Adam computer has resulted in a definite net loss for the fourth quarter, said Barbara Wruck, spokesman for Coleco. Despite reports of numerous product returns, the manufacturers of the Adam insist its returns have not exceeded 10%.

Coleco has admitted its original production projection of as many as 500,000 units in 1983 as inaccurate; actual production totals are closer to 95,000. The firm has attributed the loss to insufficient production, and states it has the production capacity in 1984 to produce approximately 100,000 units per month.

At the January Consumer Electronics Show, Coleco announced it would market a 5¼" disk drive, modem, additional 64K RAM and 170 new software titles. The firm sees IBM compatibility by year's end.

### PENNEY'S CLEARS SHELVES OF COMPUTER PRODUCTS

Product inavailability and price deterioration during the holiday rush saw J. C. Penney's clearing its shelves of computer hardware and software inventory. As of February 1, Penney's will no longer carry computer products or video games. Penney's 1984 marketing emphasis will be on its clothing lines.

The mass merchandiser carried the Texas Instruments 99/4A and Atari products in its stores. TI's exit from the Home Computer market, a shortage of Atari products and the absence of Coleco's Adam computer from store shelves left Penney's with little product to sell during the peak holiday season.

### SINCLAIR RESEARCH ANNOUNCES "QUANTUM LEAP" COMPUTER

Sinclair Research Ltd. of Britain has introduced a low-priced computer aimed at sophisticated home users, schools and businesses.

The QL computer, which stands for "quantum leap," is priced in Britain at the equivalent of $559. The QL is said to rival IBM's personal computer and Apple's IIe at a much lower price. Its price makes it comparable to Coleco's Adam.

The computer contains 128K of memory storing about 128,000 bits of information, and is sold with two tape-storage drives. A printer, video screen and additional storage devices cost extra. The product's main engine, a Motorola 68008 chip, can process 32 bits of information at a time rather than eight bits, as with more inexpensive computers. The QL's 128K stores about twice as much information as home computers.

Four software programs are included in the purchase, and enable word processing, graphic production, data processing and business planning. The QL is not compatible with either the IBM PC or Apple IIe.

Beginning late February, the QL will be sold through mail order in Britain. Retailing for $499 in the U.S., it is more of a competitor for the home computers of IBM, Commodore and Coleco. The computer offers more software and add-on parts than IBM's PCjr which, at $669, hasn't any extras. Although the Adam offers built-in writing software, printer and tape storage device, the QL's price undercuts Adam by almost $200.

### ELECTRONIC DISTRIBUTION INCREASES, SPAWNS NUMEROUS JOINT VENTURES

The relatively new concept of electronic software distribution is seeing a thorough testing, as many software manufacturers are teaming up to distribute their products to households electronically.

ActiVision and Atari have announced a joint venture to broadcast video games to households. Coleco and AT&T have also teamed up to distribute computer games, as have Rogers Cable and its partner, the Games Network. All seem to be ignoring Mattel's ill-fated Play Cable; Mattel is expected to terminate its three-year effort at electronic software distribution soon.

Atari and ActiVision will begin by broadcasting video games using the Atari 2600 and compatible machines. While compatibility expansion is planned, the duo will have approximately 14 million 2600s as its base. No facts are available as yet to management procedures and test market plans.

# SPECIAL REPORT

## CONSUMER ELECTRONICS SHOW

**By Charles La Fara**
*President, IUG*

For those who have any doubt that Texas Instruments has truly exited from the Home Computer market, those doubts could have been dispelled with a visit to the Consumer Electronics Show held January 6-10 at the Las Vegas Convention Center. The Consumer Electronics Show is the premiere format for manufacturers of consumer electronic products to show their wares to retail store buyers from around the world. Over 75,000 retail buyers attend the CES annually and plan seasonal purchases from products they see at this show.

For the last several years the Texas Instruments booth, located in the main convention center, has been the center of attention for thousands of retail buyers who wish to cash in on TI's popular Home Computer and software products. This year, however, TI's million dollar booth resembled The Day After, which was portrayed in a recent ABC television film. The only products on display at the TI booth this winter were its calculators and Learning Aids products. Gone are the days of floor-to-ceiling software racks, impressive four-channel quadrophonic sound being reproduced with TI's Home Computer, buyers with "joystick wrist" from playing too much Parsec, Munch Man or TI Invaders and also the shoulder-to-shoulder crowd of onlookers which gazed in amazement at innovative new products for the next buying season. Gone also were the backrooms and hotel suites of wondrous new prototype products such as the 99/8 and IEEE Interface cards.

Regional managers closely guarded American Express cards which just a year before had furnished entertainment to

buyers at such places as Caesar's Palace, the MGM Grand and other famous "Las Vegas Strip" nightspots. Other missing items included the familiar whine of TI's Lear jets swooping into Las Vegas' International Airport carrying corporate executives and high-dollar buyers. Instead, TI employees scrambled for last-minute hotel reservations and stand-by seats on commercial flights in and out of the City of Glitter. Even the mention of the 99/4A Home Computer inside the TI booth drew menacing glares. The ever-popular TI Hospitality party, which always included tons of fresh shrimp cocktail and caviar as well as entertainment from your favorite Coke and pudding salesman, no longer existed. This year, not even potato chip bags and Dr Pepper was offered.

What does all this mean?? Is TI ready to give up the Consumer Products business completely? Not hardly. TI still has a very strong commitment to their calculator and Learning Aids business which currently occupies a very favorable market share at the retail level. Several new products were introduced by TI, including some additions to their professional calculator line. These products, however, lack the charisma of the Home Computer.

The only TI people from the old Home Computer days attending the show were attorneys and corporate staff involved with the wind-down. During our four-day visit we had an opportunity to meet with these people and would like to share our findings with you.

## WITHDRAWAL COMMITTMENT

Texas Instruments' exit from the home computer market was reinforced by corporate attorneys and staff members attending the Consumer Electronics Show with the following statement: "We would like to exit this portion of the business as rapidly as we can and still maintain a high degree of credibility among our over two million 99/4A owners." There is, however, one problem in this statement. TI's decision to exit the Home Computer business was made without foresight and planning to do so. Thus, a large amount of confusion and frustration exists not only at the consumer level but also with some dealers and distributors. It is my personal belief that corporate leaders within the Texas Instruments organization truly desire to exit the Home Computer business with dignity and provide all known owners of their Home Computer product with a continued area of service and support for as long as they feel it is necessary.

## WARRANTY SERVICE

Those consumers who purchased their Home Computer during the third and fourth quarter of 1983 will be afforded the full one year warranty on the computer console and a 90-day warranty on all other hardware and software products produced by Texas Instruments. There is, however, a Catch-22 involved in the warranty process. Normally, a consumer would be able to simply return a defective piece of merchandise within the first 30 days to the original place of purchase for replacement. Due, however, to limited supplies of equipment and software, this is virtually impossible in many cases. We suggest that any warranty product be returned directly to regional TI service centers or to the service center in Lubbock, TX for any repair or maintenance.

## TI-CARES HELPLINE

**Texas Instruments will continue to provide for an indefinite time a toll-free help line, 1-800-TI-CARES, on a six day-per-week basis. This line is provided for 99/4A owners wishing assistance and location of regional service centers, technical assistance and availability of Texas Instruments-produced hardware and software products. In addition to providing information concerning operation and repair, the TI Helpline will also take orders for Texas Instruments products at suggested retail prices while their inventories last. Because of heavy activities on the TI-CARES Helpline, owners can expect long delays in either reaching an operator or receiving assistance. Currently, only 20% of all calls to this number are being responded to. This service is provided only to owners who cannot locate Texas Instruments products through local retail outlets.**

## CONTINUED COMMUNICATION

Texas Instruments will be taking steps to provide information to all known owners of the 99/4A Home Computer. All owners who have received copies of TI's Home Computer Newsletter or direct mail pieces from Texas Instruments will continue to receive information concerning their computer directly from TI. Additionally, all new owners who send in the Owner Information cards will be added to this mail list. Any owners of a 99/4 or 99/4A Home Computer who is currently not receiving information from Texas Instruments can do so by sending a post card with their name, address, and computer serial number to Texas Instruments, P.O. Box 53, Lubbock, TX 79408. Current plans call for the publication of at least two more TI Home Computer Newsletters to be produced in January and April 1984. In addition, arrangements have been made for quarterly mailings for interested third party hardware and software manufacturers which should continue to keep 99/4A owners informed of new developments from third party hardware and software manufacturers.

## THIRD PARTY SOFTWARE AND PERIPHERALS

On January 9, 1984 Texas Instruments announced the signing of software agreements with Walt Disney, Sierra On-Line, and Imagic for the responsibility of manufacturing and marketing 99/4A software products by these third party producers.

"This agreement is part of our plan to provide alternative sources of quality software for owners of TI 99/4A Home Computers," said Dale Osborn, manager in charge of the phase-out of TI's Home Computer operations. "This is part of TI's commitment to its consumers to take steps to provide continued support for 99/4A owners," Osborn went on to say.

In addition to Sierra On-Line and Imagic, TI continues to discuss with third parties the manufacturing and marketing of additional software packages which have previously been produced only by Texas Instruments, Incorporated.

**TI also announced that it will issue licenses for its Auto-Incrementing GROM chip to interested third party manufacturers. This is a total reversal of its unwillingness to work with third party manufacturers as short as six months ago. (See Enthusiast '99, Vol. 1, No. 2, July 1983, p.4.) Current TI plans also call for additional manufacturing of certain TI hardware and software products until all piece part inventories are depleted or sold to interested third party manufacturers. Our opinion is that unless Texas Instruments acts rapidly in the area of solving software shortage problems, many new owners will become frustrated to the point of not using their machines. We also feel that TI should have a strong moral committment to keeping its 99/4A owners well advised in the area of hardware and software availability.**

Hundreds of questions as to how long hardware and software products will remain available and from what sources it will come still remain a mystery to us all, and it may be several months until all of the pieces to this intricate puzzle are pieced together. We here at the International 99/4 Users-Group will continue to provide all our members with as much information as possible on the supply of hardware and software products and offer as much assistance as we can to our members during this crisis period. We cannot stress the importance of PATIENCE during these trying times. Providing a reliable service to our members has been made extremely difficult; however, we are doing the best we can. It is extremely difficult for the IUG to respond immediately when we do not control manufacturing and total distribution of hardware and software products which have in the past been made readily available to us.

These are extremely trying times for all owners of the 99/4A Home Computer and it is our consensus and opinion that unless Texas Instruments develops structured plans for their withdrawal from the Home Computer market and acts on them immediately, confusion and frustration will abound for many months to come.

# VIEWS

## Enthusiast '99™

OFFICIAL PUBLICATION OF THE INTERNATIONAL 99/4 USERS GROUP

It appears that the International 99/4 Users-Group is in the right place at the right time. As a result of Texas Instruments' withdrawal from the Home Computer market, people are joining the Users-Group at a record pace, just in time to receive this issue of Enthusiast '99.

This issue contains useful information concerning the present third party market. An attempt was made to cover each area of interest to the consumer from educational to business to entertainment. In addition, some excellent hardware products are also featured, along with programming helps from A Woman's View, Et Cetera and The Assembly Line.

Third party producers have come to realize that a positive and extensive market can be reached through the IUG. In addition, their products must be of top quality, for they now serve a specific and serious group of consumers; IUG members.

The International 99/4 Users-Group is continuing in its quest to become a more service-oriented organization, and to gain memberships from all known owners of the TI 99/4 and 99/4A Home Computer. This would result in a phenomenal increase in programs in the Software Exchange Library, and keep Enthusiast '99 a prime medium of exposure for third party producers. A high volume of members will serve to keep product prices at reasonable levels, and tap a huge reservoir of unshared ideas and discoveries. All of these factors will help to keep the IUG a powerful organization--and you, our members, are at the core of its success.

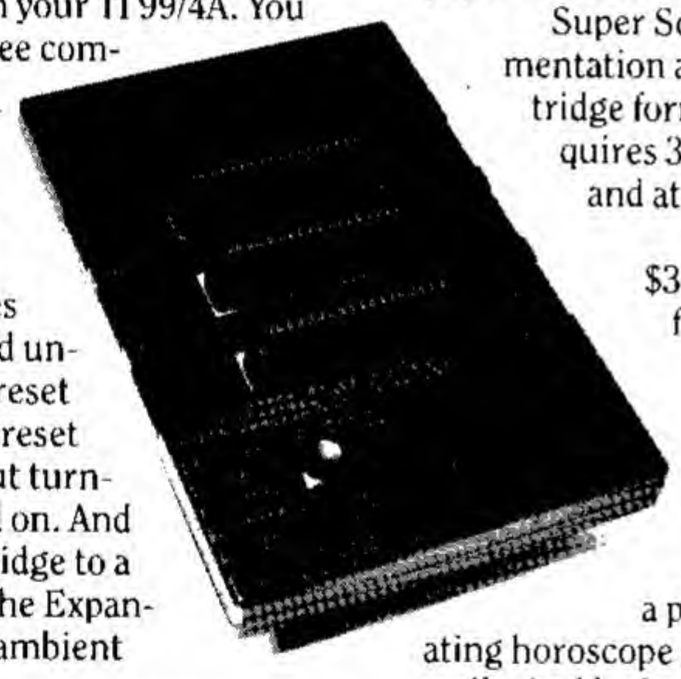
Dana

# Where do you go when there's nowhere to go? Navarone.

Things look pretty bad for TI 99/4A owners. So what happens now that your Console can't go home again? Read on. You'll see what we have up our sleeves.

## GIVE YOUR CONSOLE ETERNAL LIFE

We have taken our successful *Widgit* and made it better. Introducing our *New Cartridge Expander.* It inserts directly into the game slot on your TI 99/4A. You can plug in up to three command modules at one time. With three sockets and a selector switch, you can choose between your favorite modules without plugging and unplugging. A built-in reset button allows you to reset the computer without turning the power off and on. And by bringing the cartridge to a verticle position on the Expander, we've increased ambient cooling.



Not only have we eliminated overheating and program failure, we have given your Console longer life. By removing the possibility of wearing out the Console's cartridge slot, we've just about made your TI immortal. After all, when something wears out or breaks it won't be easy to find service or parts anymore.

The Cartridge Expander is easy to use and easy on your pocket book—priced at a low $39.95. When ordering, ask for catalog number WD-01.

## ELECTRIFY YOUR FILES

Now you can turn those paper files into electronic files. And get any information you want, in any order you want, faster.

We proudly present our *Data Base Management Series,* featuring many programs to increase your personal productivity. Now you can spend less time searching and more time doing.

*Super Sort,* our assembly language cartridge program provides many DBM functions. It is menu driven to arrange your files in ascending or descending order with up to six nested keys. And is a high speed disk sort utility program written to handle large files. It is limited only by the disk space available, so you can shrink your paper files down to a manageable size.

Super Sort comes with full documentation and is available in cartridge format only. The program requires 32K memory expansion and at least one disk drive.

Find more for less, only $39.95. When ordering, ask for catalog number AS-03.

## TOP OF THE CHARTS

Within *Astrology-Horoscope Maker* we have written a powerful program for creating horoscope charts easily. And look at the kind of chart you are able to print.

This software program determines the precise location of each planet automatically. You won't need tables and lengthy calculations. You are able to print full wheel charts selecting up to nine house systems and have the planets appear in their respective houses. Included are elements, declinations, dignitaries and more.

Astrology-Horoscope Maker is so easy to use, too. Just answer the questions in the menu and you'll have it. But, don't think this uncomplicated program is a mere game. It is written for the professional astrologer and student of astrology. And your charts may be saved on a diskette for future printing.

You will need a drive as this is only available on diskette. The program requires Extended Basic, 32K memory expansion and an 80 col. printer such as Epson 80 or Okidata 82A. Priced at only $49.95 it is real chart topper. When ordering, ask for catalog number EB-03.

## ESTABLISH ORDER

Our *Homework Helper* is a powerful organizational tool inside a cartridge. Just plug it in for a completely menu driven program that's easy to use and a big help with homework.

Your child can keep record of assignments. Or use the provided book report formats. Or generate problem worksheets. A practical applications way to teach or learn computer literacy.

Homework Helper is available in cartridge only, requires 32K memory expansion and at least one disk drive. And it is priced at only $39.95. When ordering, ask for AS-04.

VISA and MasterCard welcome. To order by phone, call (408) 866-8579. If ordering by mail, send Cashier's check or Money Order. (Personal checks take 10 days to clear.) C.O.D.'s add $5.00.

Please state items you are ordering, quantity and the price of each item. Allow 4 to 6 weeks delivery. Add 5% of total order for shipping and handling. CA residents add 6.5% sales tax.

For more information on other Navarone products request our catalog.

Dealer inquiries welcome.

NAVARONE

Navarone Industries, 510 Lawrence Expressway, #800, Sunnyvale, CA 94086

# MEMBER SPOTLIGHT


## STEVE SALLEE

4801 Selwyn Dr.
Winston-Salem, NC 27104


From the time I was a small child, I have been fascinated by pushbuttons, switches, knobs and the electronic circuits controlled by them. Maybe this attraction is due to the sense of power one gets when the touch of a finger can cause a dramatic response such as flooding a room with light, activating a motor, or filling a computer screen with characters and graphics. Most people take these things for granted, but the ability to manipulate the world using pushbuttons took on special meaning to me in the summer of 1962. Left paralyzed from the shoulders down after I came out second best in a battle with the bottom of a swimming pool in Colorado, my spinal cord injury left me with limited arm movement, and unfortunately I lost the ability to move individual fingers.



Fortunately, through help from rehabilitation therapists, my family, and my fiance (soon to be wife) Athene, I was able to return to my college studies in biological science at Oklahoma University south of Oklahoma City at Norman.

College life in a wheelchair can be an exciting challenge. There were frustrations, such as not being able to find a ramp into a building after driving there with hand controls and laboriously unloading your chair and dragging yourself into it. Of course, humorous events keep spirits high — such as the time a friend's electric wheelchair control got stuck, causing him to race around in tight circles for several minutes before being rescued by a passerby.

After receiving a BS and Masters degree, I was fortunate to obtain a college biology teaching position first in Kansas and later in North Carolina. After teaching and conducting research for the last fifteen years, I have grown even more dependent upon pushbuttons for a fuller and happier life. I push a button to open my garage door and to control appliances. I also operate my TV, video recorder, and house lights from my bed by remote control. However, my favorite buttons are on my ham radio and computer equipment.

Our family's life was changed rather dramatically almost four years ago when I took my wife and son Mark into a new computer store to examine the selection of electronic marvels called home computers, which contained all those irresistible buttons. I had already decided that we just could not live without one and had entered the store hoping to get the new Apple computer I had been reading about; however, tucked between the Apple and CBM displays was a sleek little machine with the strange label TI-99/4. All the computer demonstrations were good, but I kept being drawn back to the little Texas Instruments unit. The graphics were outstanding and the sixteen bit microprocessor was bound to be the wave of the future. When my six-year old plugged in the Beginning Grammar module and started "playing games" with nouns, verbs, and other parts of speech, I was hooked for sure.

The editing features of the TI-99/4 were so easy to use that my son had figured them out in a couple of minutes, while the computer salesman had trouble just explaining them on the Apple. The auto numbering, resequencing, trace, and other error handling functions of the TI were also easy to use. The prepackaged programs in modules seemed the ideal way to avoid tape or disk problems that small children were likely to encounter. All in all, the TI-99/4 seemed to be the best unit for our whole family; easy to operate for Mark, yet powerful enough to do the biostatistics that I needed. As a bonus, the game graphics were enough to challenge the most demanding player.

After paying about $1,200 for the console, color monitor, and a few modules, I was surprised to find that we were getting the first one sold by that store and one of the first in the state of North Carolina. The hardest part was waiting the three months before the unit would be delivered. The time was spent reading all I could find on BASIC programming.

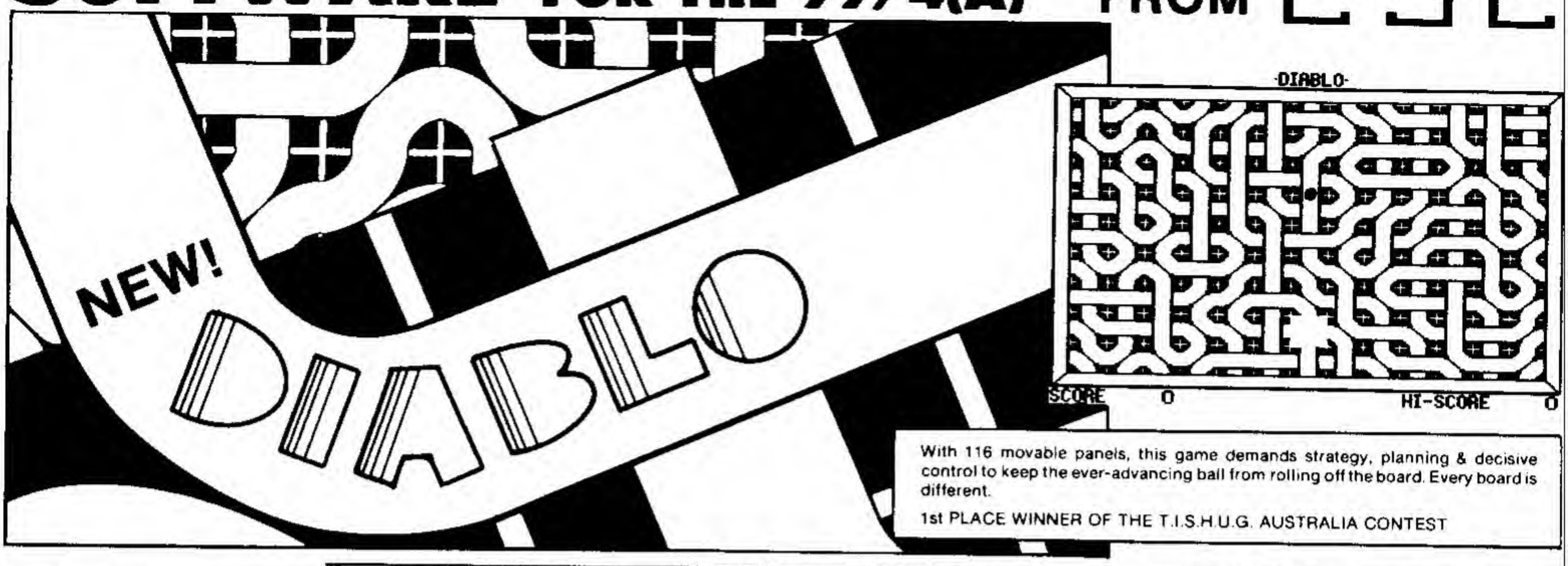
Finally the unit arrived and was adopted into the family. The only problem was that the cassette interface cable was backordered and would not be available for some time. This came to be one of a long series of items that were ordered from TI but took ages to arrive. It was very frustrating to type in a program and lose it when the computer was turned off, especially considering that I type with a fat, first-grade pencil with the eraser end hitting the keys and the other hand touching the shift key. It was a day of rejoicing when the cable finally arrived.

The next item added to our "computer room" was an old printout unit from a finance company that had entered the surplus market. I was able to trade an old piece of ham radio gear for it at a local hamfest. It had to have a new set of keytops, an ASCII print ball, a board to convert it to RS232, a cable, and finally a new coat of black paint before it was fully adopted into our family. My son had named the computer Alvin so he called the printer Albert. Soon after this, Alvin got a voice. The speech synthesizer had arrived, and added much to our programs.

It was very lonely being an early TI-99/4 owner. Over a year passed before I met someone else with the same computer, so I was overjoyed to get a letter from Charles La Fara dated August 12, 1980 stating that a "nation-wide 99/4 Home Computer Users-Group" was being formed for the exchange of programs and to provide technical and programming assistance. I sent back the reply card that day and have been a satisfied member ever since. I had the pleasure of meeting Charlie in person in Oklahoma and he was very helpful in suggesting ways to begin a local Users Group for the Winston-Salem, NC area. He even had his computer give me a list of North Carolina 99ers. I am now president of the Forsyth 99'er Computer User's Group and editor of our newsletter

(continued on page 43)

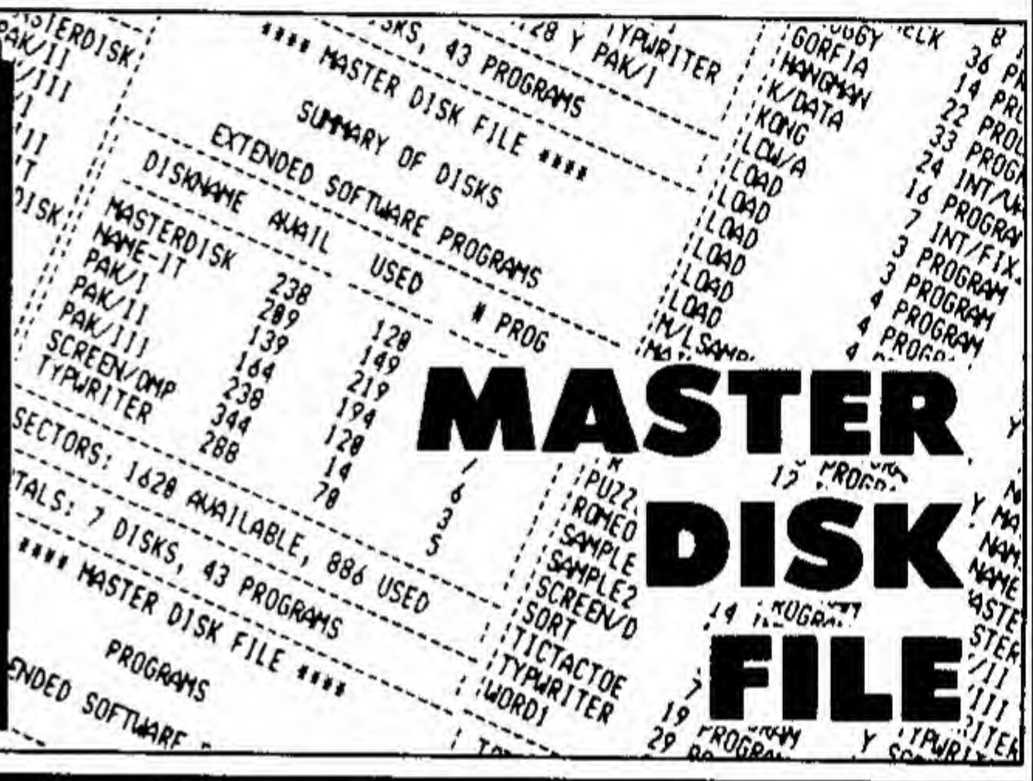
# SOFTWARE FOR THE 99/4(A) FROM EXTENDED SOFTWARE COMPANY




With 116 movable panels, this game demands strategy, planning & decisive control to keep the ever-advancing ball from rolling off the board. Every board is different.

1st PLACE WINNER OF THE T.I.S.H.U.G. AUSTRALIA CONTEST

**MASTER DISK FILE** — *A master index of your disks.*
MASTER DISK FILE uses a large portion of the 90K bytes available on a single sided disk as virtual memory to create and store a perpetual file of disks, programs and applicable data. Data can be added, removed or updated. The file may then be accessed for viewing on the screen or for printing several types of lists on a printer.
Insert disk into drive, it is catalogued then can be filed.
Reads up to 95 programs and files per disk.
Maximum of 120 disks or 1100 programs may be filed.
Does not require memory expansion.
Supports single or double sided disks (or mixed).
Supports single or multiple drives.
Programs are catalogued from disk in order and merged into the file; no sort is required after filing.
SEARCH for disks or programs by name.
Look-up time from a cold start: under one minute; from a running program: 15 to 25 seconds!
List on screen or a printer in alphabetical order by program name or disk name.
Up-date disks that have been changed by inserting into drive. Automatically replaces and up-dates old information.
May be used without a printer (on screen); file is portable and can be taken to a friend's for printing.
Supports any printer: serial or parallel.
Diskette (only) ..... $15.00
GUARANTEED TO BE THE BEST DISK CATALOG UTILITY FOR THE 99/4A




## GAMES PAK/III

**KONG** - must fight his way past barrels, ladders & trap doors to save Roxanne from Igor's bomb. Six different screens. Joystick req. ..... $15.00

**BOUNCER** — bounds from one trampoline to another, scoring points while avoiding the arrows which will burst him. 6 screens. Joysticks req. ..... $15.00

**ROMEO** — must traverse a desert, stream (alligators & sharks) & cavern to obtain a fitting reward. Joysticks req. ..... $15.00

## GAMES PAK/II

**ARTILLERY** — The opposing force must be destroyed by determining angle and force of each shot. An ever-changing wind complicates matters. 1 or 2 players ..... $9.95

**DE-CYPHER** — An encrypted message is displayed and guesses change all corresponding letters to the guess. Comes with 50 messages which can be changed .... $9.95

**PUZZLE 15** — Slide alphabetic squares into order. Multiple squares may be moved. Computer keeps track of moves & has replay feature ..... $9.95

**FLIP CHECKERS** — Sandwich opponent's piece to flip it to your color. 1 or 2 players. Real-time clock. Fast computer decisons. Joysticks req. ..... $9.95

## GAMES PAK/I

**FROGGY** — Jump FROGGY across 10 lanes of traffic, then across 6 logs. Keyboard or joysticks ..... $9.95

**EXTENDED BASEBALL** - Joystick control of the pitcher & batter, and individual batting averages. Multi-base & multi-runner plays. Joysticks req. ..... $9.95

**GORFIA PESTULITIS** - Joystick control of a laser sight or inertia influenced space mines to shoot down the invading Gorfians. Joysticks req. ..... $9.95

**EXTENDED HANGMAN** - Quick graphics, music, color, speech (optional) and sound are added to keep the players entertained. Includes 580 words of 4 to 9 letters in length in easy, medium and difficult groups ..... $9.95

**TIC-TAC-TOE** — Quick set-up and quick decision making at four levels of difficulty. The levels avoid the frustration of the novice never having a chance to win ..... $9.95

**TYPWRITER**
*A 16K WORD PROCESSOR on Cassette or Disk*
*Any* Input/Output storage of text — disk, cassette, cassette input/disk output, or vice versa.
Selectable *right justify*.
Complete text *Editing* — by cursor control; including insert & delete lines, partial print, printer halt or abort without text loss, page FWD & BKWD, and more.
Complete *Software Control* of Printer (depending upon its capabilities) — for enhanced print, underlining, formatting, 28 to 254 characters per print line, etc.
*No Special Equipment* — monitor, console, Extended Basic module, C or D, printer.
Comes with a 20 page instruction booklet.
Cassette $32.00 Diskette $35.00

**SCREEN/DUMP** — Print the screen to a dot-addressable, 8-bit printer. Does not require extra memory! Disk version is simple to use. Cassette version requires mild programming knowledge ..... $12.00

**NAME-IT**
*DATA BASE* for: *Mail Lists, Labels, Files*
*Records*: 250 records per diskette consisting of up to nine 28-character items per record.
Prompts: user designated prompts.
Complete *File Sort*: 250 records in *100 Seconds*.
Search; Pre-set; print labels & lists.
Includes a FORM LETTER program that uses NAME-IT data in TYPWRITER generated form letters
Cassette version differs from disk version.
Cassette $32.00 Diskette $35.00

NOTE: Should you decide to up-grade to the TI-WRITER module, TYPWRITER and NAME-IT data can be converted for use by that module. NAME-IT alone, will generate 250 TI-WRITER form letter records.
TI-WRITER is copyrighted software of Texas Instr.

A detailed catalog is available free. Circle "FREE" on the order form or send a letter or postcard.

## ORDER FORM

EXTENDED BASIC MODULE REQUIRED FOR ALL PROGRAMS

| | | | |
|---|---|---|---|
| **CATALOG** | FREE | | $ FREE |
| **GAMES PAK/I** (Froggy, Extended Baseball, Gorfia Pestulitis, Extended Hangman, Tic-Tac-Toe) | $26.95 | (C or D) | $ |
| **GAMES PAK/II** (Artillery, De-Cypher, Puzzle 15, Flip Checkers) | $26.95 | (C or D) | $ |
| **GAMES PAK/III** (Kong, Bouncer, Romeo) | $26.95 | (C or D) | $ |
| **TYPWRITER** (word processor) | $32.00 | (C price) | $ |
| | $35.00 | (D price) | $ |
| **NAME-IT** (data base/mail list) | $32.00 | (C price) | $ |
| | $35.00 | (D price) | $ |
| **SCREEN/DUMP** (printer required) | $12.00 | (C or D) | $ |
| **MASTER DISK FILE** | $15.00 | (D only) | $ |
| **INDIVIDUAL GAMES:** | | | |
| **DIABLO** | $19.95 | (C or D) | $ |

☐ Froggy ☐ Extended Baseball ☐ Gorfia Pestulitis ☐ Extended Hangman ☐ Bouncer ☐ Kong
☐ Tic-Tac-Toe ☐ Artillery ☐ De-Cypher ☐ Puzzle 15 ☐ Romeo ☐ Flip Checkers

Total Individual Games: (C or D) $ _____
Shipping & Handling (1st Class Mail): Included
Sales Tax: Included
Add $2.00 if C.O.D. (U.S. Mail Only): $ _____
Check or money order or C.O.D. Total: $ _____
PLEASE SPECIFY CASSETTE OR DISK

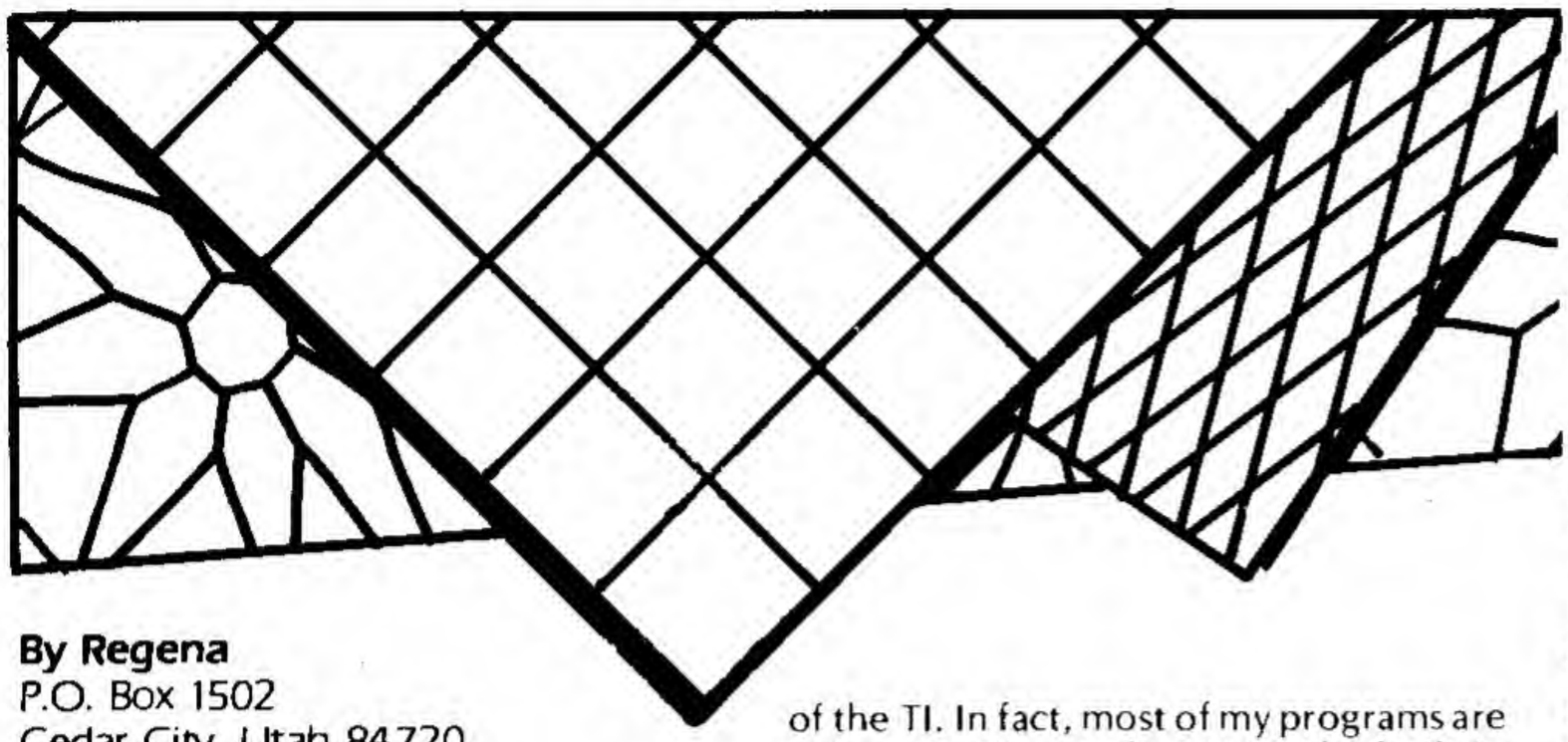
Send this form or a substitute with check or money order to:
**Extended Software Company**
11987 Cedarcreek Drive
Cincinnati, Ohio 45240
(513) 825-6645

**SATISFACTION GUARANTEED**
IF YOU ARE NOT COMPLETELY SATISFIELD, YOU MAY RETURN THE PROGRAMS (and instructions) WITHIN 15 DAYS FOR A FULL REFUND OF YOUR PURCHASE PRICE.

Dealer and programmer inquiries welcome.

Cut Here (may be copied or substituted)

# A WOMAN'S VIEW



**By Regena**
P.O. Box 1502
Cedar City, Utah 84720

Greetings for the New Year! And welcome to all our new members who may have just received a TI-99/4A for Christmas.

In answer to all the phone calls and letters I have been getting since Texas Instrument's announcement concerning home computers, I *do* intend to keep writing for the TI-99/4 and TI-99/4A, and I'll keep publishing in magazines and books as long as my publishers feel there is a significant reader base for the TI. Of course, I also publish articles and programs for other home computers so the TI isn't my only source of income.

I still believe the TI is a great home computer. Contrary to many recent buyer's guides, you do *not* have to have the "expensive peripherals" nor an "expensive video monitor" to enjoy the features of the TI. In fact, most of my programs are written in TI console BASIC (the built-in BASIC) and can be used with only the computer hooked up to a regular color television set and a cassette recorder to store the program. All of my programs are written without memory expansion — 16K is not that limiting for many programs because our "nonstandard BASIC" is really quite powerful. Akggh! — enough soapboxing! I really wanted a new computer for my birthday in November, but I ended up buying several more TI's!

I do foresee one major problem. In many of my computer presentations I use a TI-99/4A to show the audience what a **computer is and what a variety of things a** computer can do. I use programs on the TI to illustrate what good educational software is and how color and graphics can enhance an educational program. Of course one of the main features is how easily a child can load a program by simply choosing a plug-in module. The problem is how can I answer when someone asks, "Where can I buy one of these computers?" Well, umm . . . the TI is no longer available . . . .

I do think that those of us who have TI computers will continue to enjoy them for many years. I think you'll find that this International 99/4 Users-Group will be a main source of continued information and service. Local users groups will also be important sources of support. I have found that other TI owners are very friendly and helpful and willing to share information.

My program for this issue is "Quilt Squares". I have always admired patchwork quilts, but my quilting projects have ended up as small baby quilts or even doll quilts because I soon tired of the tedious task of cutting and piecing little squares and triangles. This program allows you to design a basic quilting square, then the computer will expand the pattern so you can see how it would look over a larger area.

When you RUN the program, the first question you are asked is "How many colors?" You may choose 2, 3 or 4 colors for your quilt. Ten color squares are then shown, and you may choose your quilt colors by pressing the number above the color you want. The screen clears and the possible quilt squares are shown at the right of the screen. You may have either a square of one color or a square made up of two triangles of color. At the left of the screen is a grid of four squares by four squares. One by one a position will blink. Use the arrow keys to move up, down, left, or right and position the arrow next to the type of quilt square you want, then press the ENTER key. That pattern will then appear on the basic quilt square grid. Next you will be given a chance to change that pattern — press 1 if that square is okay and 2 if you want to change. Continue this process for the sixteen squares. The computer will then reproduce your pattern so you can see how six of your basic squares would look placed together. When you are finished admiring your quick quilting work, you may press 1 to try a new pattern or 2 to end the program.

Since DATA statements are the main source of typing errors in programs, I decided to try this program without any DATA statements. However, as you are keying in this program you will need to be careful where there are several similar statements together or where there are groups of similar statements and you could accidentally skip a few statements.

**Since DATA statements are the main source of typing errors in programs, I decided to try this program without any DATA statements.**

Lines 150-300 define the graphics characters. Characters 91-94 draw the basic grid, and Characters 96-99 are used to blink one square red. Lines 230-300 define the various quilt squares in different color sets — solid squares and the four possible patterns of triangles for two colors. Lines 310-350 define the colors so you can choose the colors for your quilt design. The color squares are printed in Lines 410-510.

Lines 520-650 establish the quilt colors, and the color numbers are in the C( ) array. Lines 2060-2210 contain a subroutine detecting which number is pressed for the color desired, then A is the color number.

Lines 660-700 clear the screen and print the basic quilt grid plus the options to change a square. Lines 710-770 define the colors for the quilt depending on what you have chosen. Lines 780-1130 draw the little quilt patterns at the right of the screen which represent the possible squares you may choose. N is the number of colors you have chosen for the quilt. RX and RY are row and column coordinates used later as limits for the blinking arrow while you are choosing squares.

Lines 1140-1820 are the main loop for designing the basic quilt square. PX and PY are the row and column coordinates in the larger basic block, and the FOR-NEXT loops move from square to square as choices are made. Lines 1160-1190 draw the red square to indicate which position you are drawing. TX and TY are the coordinates for the arrow which moves as you press the arrow keys to indicate which quilt pattern you want. Lines 1220-1470 move the little arrow within the limits of RX and RY.

---

***I do think that those of us who have TI computers will continue to enjoy them for many years.***

---

Call GCHAR is a function that returns the character number of what is currently on the screen at the position specified. Line 1480 CALL GCHAR(TX,TY+1,G) says to look at row TX and column TY+1 to get the character number G of the quilt pattern desired. Line 1490 uses a little arithmetic to figure out which graphics character is involved, then Lines 1500-1740 draw that pattern on the larger basic grid. Each square on the basic grid is actually made up of four graphics squares.

Lines 1750-1800 erase the arrow at the quilt patterns and ask if the pattern you have just drawn is okay or if you wish to change. If you want to change, the program transfers back to Line 1160 and the process of choosing is repeated at the same basic grid position. If the square is okay, the program goes to the next square in the basic grid.

After all sixteen squares have been designed, Lines 1850-1940 repeat the pattern over more of the screen. The FOR-NEXT loops move from square to square of the basic grid. Line 1870 CALL GCHAR(PX,PY,G) gets the character number in each row and column position of the basic grid, then Lines 1880-1920 repeat that character at five other positions on the screen to show the repeating pattern on the quilt.

Lines 1950-2050 present the option to try a new pattern or stop (end) this program and the program branches appropriately.

Until next issue — I hope you have fun with this new type of computerized quilting. *(printout on page 48)*

# LIBRARY CORNER

# THE FEAR OF FILING — PART II

**By Guy S. Romano**
Senior Staff Editor
116 Carl St.
San Francisco, CA 94117
(415) 753-1194

To continue on with our discussion of data files for use with our computer, we need to take a good look at two key words of great significance. The words are SEQUENTIAL and RANDOM. Again, they can be dangerous because we are dealing with regular English words that may have a very specific and different meaning when applied to the world of computer jargon.

Let's look first at the word SEQUENTIAL and what it will mean for us in the computer sense. Does anyone remember when one of the hot features on the "new" Thunderbird back in the 60's was "sequential directional signals"? They reminded me of one of those large blinking arrow signs ending with "Joe's Cafe" that one saw on old highways. Well they lit up one after the other to make a moving arrow to show the direction. That's the key — "one after the other" AND each one right after the preceding one. If you have ever seen that commercial on television then you know how Sandy Duncan eats her Wheat Thins — sequentially!

When we read a book, we read it sequentially, starting at page one and then continuing on through the book, one page after the other. This sequential business represents a kind of orderliness to our lives and gives security in that we can anticipate what should come next. Most of what we do in our daily lives is structured sequentially. We go to school sequentially from Kindergarten through college. We do not jump around going first to grade 12 and then grade one.

Sequential type files are the most commonly used type in data file storage. This is evidenced by the fact that "SEQUENTIAL" is a default setting in opening a file so that you do not even have to include it when setting up your file attributes. Remember that UPDATE, SEQUENTIAL, VARIABLE, attributes are all default settings since they are the most commonly used. Sequential files are usually best for data files that can be considered as "batch" records. This includes text files for a word-processing program or files in which all of the records will be read into the computer all at once and then have the data stored in arrays in the program for faster access. The fact that sequential files can be VARIABLE in length means that they will take up less storage space on disk or tape and therefore can be saved or loaded faster. Also, we do not have to be very careful about the length of records since the computer takes care of that management aspect. If we want to create sequential files with records of FIXED length we must first analyze each field of the proposed records to determine the total length of the record. This is done by adding one to the maximum length of each field and then adding it all together.

In practical terms, if we were to follow this format with the previous primitive examples of our "database" program, we will have to figure out the longest name in our list, the longest address, etc. Let's say that the longest name we have is 27 characters long, the longest address is 28 and the birthday will always be 8 characters long (11/05/83). We then add those together to get 63 AND we must add one more because the computer will store the length of that record in that extra character. Consequently, we will open our file with FIXED 64 as its attribute. If we could actually look at a FIXED record in our file it would look like this:

```
JOHN SMITHØ123 OAK STRE
ETØØ6/15/55ØØØØØØØØØØØ
ØØØØØØØØØØØØØØØØØØ
```

The computer fills in those zeros (30) to make each record the same length. If in entering data we entered a record that was longer than what we first established for the record length, the computer would either chop off any excess, or worse, write the leftover on the beginning of the next record. Our files would become an unreadable mess. Remember that comment about *order and discipline*. The computer will always expect that data be in the same order and in the same place within a record. If you order it to do one thing and then give it another it will balk and give you garbage quite dutifully.

Now take that word "RANDOM". It reminds us of "casual", "helter-skelter", "willy-nilly" and other words like that. Well, the difference in meaning between real world English and Computerese with this word exhibits enough differing aspects for it to qualify as a one-word oxymoron. "Random" in TI Basic, is described perhaps more correctly as RELATIVE. A better word because the ability to *seem* to search for data in a random fashion is always *related* to something else. There is a much more rigid discipline to RELATIVE files than to sequential ones because each record must be the same length and with the data for that record in the very same order all the time, otherwise the computer will not be able to find anything.

You have probably heard diskettes referred to as "mass storage random access modules". You may look upon that phrase with trepidation but it is really nothing new at all. Only the fancy terminology is "new". The concept is as old as mankind. For example, I used a word back there that may not be familiar to you. If you want to know that it means you go to a mass storage random access module, usually with the title "Webster's", and go about your search. HOW you go about that search is exactly the same way the computer looks for RELATIVE records. Stop and walk through the process mentally. Notice that you do not start at the first page of the dictionary and go through one page at a time (sequentially) until you arrive at the word. You take a shortcut and use a "pointer array" to jump directly to the section you want. In this case the pointer array is composed of those thumb tabs in a dictionary showing where various alphabetic sections are located. Then what happens? We do not go to "O" and read through all the listing. Rather we use a second pointer array, the listing in the upper left hand corner of each page that shows us the *range* of words on each page. Thus you see that what we are actually doing is going from a coarse (pick up the dictionary) to ever finer tuning to locate what we want (finding the actual word). This is exactly what the computer does in a record search. To draw an even better analogy, let's look at an even more complete mass storage random access module — the Bible. Historically, a monk in the sixth century was deeply concerned about the complexity of finding things in the Bible. So he created what is now called a "Concordance". (If that man were alive today he would probably be the president of IBM.) Anyway, because of his work I can say to you, "Matthew 7:7" and you can jump directly to the biblical passage.

This same system is also used in Judaism so if you want to find something in the Tenach you go about it in the same way. As an example, let us say we want to find a passage in the Bible or the Tenach that discusses peace. We go the the Concor-

# CP/M®

## PROCESSOR FOR THE TEXAS INSTRUMENTS 99/4A

# NOW YOUR TI-99/4A REALLY MEANS BUSINESS!

*Morning Star Software introduces the CP/M Processor for the TI-99/4A. Now the advantages of the most accepted business programs are yours. With the CP/M Processor, your TI-99/4A is ready to enter the business world.*

*The Morning Star Software CP/M Processor fits into your peripheral expansion system like any expansion card. It's as simple as that. Once the CP/M Processor is activated, you have all the advantages of two processors: The CP/M Processor handles all the program computations, while the TI Processor channels the information to the screen, disks and printers. This gives you the flexibility you've always wanted.*

*CP/M is the industry standard business format for computers today. CP/M offers you the advantage of working with programs that were designed to meet the needs of the professional business world. Powerful commands, time saving functions and proven methods are the result.*

*The CP/M Processor by Morning Star is built to take advantage of both the TI computer and CP/M technology. The CP/M Processor contains a 5MHz 8085 CPU which, with the 16-bit TI-9900 CPU, translates to tremendous processing power. Your TI-99/4A combined with the CP/M Processor becomes a true business tool at a fraction of the expense of a new computer system.*

- *Needs no modifications to the TI-99/4A system.*
- *64K RAM, 8K ROM.*
- *Makes additional use of the TI 32K memory expansion.*
- *Supplemental installation and usage manual included.*
- *Contains sufficient memory for virtually all programs.*
- *Compatible with TI-99/4A storage diskettes.*
- *Provides 60K transient program area.*
- *Fully burned-in and test cycled to ensure reliability.*
- *Fully shielded and protective metal enclosure.*
- *Ultra-reliable gold connector assembly.*
- *No chip sockets for better reliability.*
- *90-day warranty.*
- *CP/M-80 2.2 disk-based operating system included.*
- *Digital Research™ CP/M user's manual included.*
- *Access to more than 10,000 commercially available software packages.*

***THE MORNING STAR CP/M PROCESSOR***

# $595

*TO ORDER: Check, Money Order, Master Card, VISA, and C.O.D. orders accepted. Add $5.00 shipping and handling. Dealer inquiries welcomed.*

## COMPATIBLE CP/M® SOFTWARE

*As Morning Star reviews CP/M packages, we will continue to update our software selection. The programs we offer are the products which demonstrate outstanding values. They have met our criteria of being extremely user friendly, comprehensive in content, plus make the best use of the CP/M card's capabilities. They also represent the tasks most needed to be covered by a business computer.*

***SUPERWRITER™ by Sorcim***

*This is the word processing program you've been waiting for. Designed to be easily understood in ten minutes with straightforward language and screen prompts. Written specifically for the business person with word processing requirements. It provides all key features of a professional text editor. It incorporates form letters, report generation, list maintenance and even a spelling checker, the award winning SpellGuard™. Similar features cost as much as $900. SuperWriter™ only* **$295**

***PERSONAL PEARL™ Data-Base Manager***

*Now easily create data systems for business, accounting, marketing, office management, education, and more. A complete user's manual, tutorial disk and screen menu guides even novices through the most sophisticated data base manager available. Unlimited files, forms and reports can be created.* ***Info World*** *reported "Personal Pearl offers you the strongest, easiest to use data-base manager I have seen at any price".* **$295**

***THE RANDOM HOUSE PROOF READER***

*20,000 word Random House dictionary. Displays spelling errors in context and can make spelling suggestions. You edit the word. Corrections are automatically checked and substituted into the document. File new words into the dictionary with a keystroke.* **$50**

***CBASIC® INTERPRETER***

*The language of choice for many CP/M compatible software packages. This brings you a variety of established and time tested applications and allows you to write your own CP/M compatible programs* **$150**

***BUSINESS MASTER® PLUS***

*A comprehensive business package designed to meet the needs of leading businesses. The package of 10 diskettes includes General Ledger, Payroll, Fixed Assets Accounting, Accounts Receivable, Accounts Payable, Inventory, and Mailing List programs. Requires CBASIC Interpreter.* **$289**

# MORNING STAR SOFTWARE

*4325 S.W. 109th Ave., Beaverton, Oregon 97005. Toll-Free 1-800-824-2412, in Oregon 503/646-4695, TEX NET NO. TI3416*

*Peripheral Requirements: Peripheral Expansion System, disk, disk controller 32K RAM card.*

dance and look up "peace". It then lists all the passages with that word in it. We can now jump to those passages in the Bible directly in what *seems* to be a "random" fashion. In reality we were able to make these direct jumps only RELATIVE to the various pointer arrays. It would appear that we are using a rather circuitous route to find something in a computer record search by first looking something up in one record so that we can then find it in another. But in this instance the most direct and efficient route is NOT the shortest route. Dare I venture to interject, "The shortest (fastest) distance between two points is not the straight line!"?

Here you begin to see that in data files in which you must do a lot of looking up of individual records all the time, you must use RELATIVE files and cassette storage becomes worthless. You simply do not have the time to start reading a file at the beginning each time and read through it sequentially until you find what you are looking for. In any program that requires random search capability you must have a *search* of records. They can only be accessed sequentially. Remember I said "accessed"! You can create a RELATIVE file on cassette tape. But you cannot *access* those records randomly. Think of what would happen if we could not have relative files. You want to have your current balance checked at your bank. Your family name is Zyxux. Are you going to wait while the computer starts with "Abab" and looks through each record until it gets to your name? Hardly! That would be intolerable and would make a "random access" Rolodex a welcome sight.

As I said before, relative files are necessarily more complex by nature. More care must be taken in our keeping track of things if we do not have those pointer arrays to help us. A discussion of the various levels (hierarchical structures) of search files (pointer arrays) is far beyond the scope of this article. So let us continue with a look at the most primitive level of relative file.

Look at the way I have changed the lines of the program from the first part of this article to reflect use of RELATIVE files. First off, you notice that reference to cassette have been eliminated.

```
 10 CALL CLEAR
 80 A$="DSK1."
 90 INPUT "WHAT IS THE N
    AME OF THIS FI
    LE? ":B$
100 B$=SEG$(B$,1,10)
110 A$=A$&B$
120 INPUT "WHAT IS THE L
    ENGTH OF EACH RECO
    RD? ":A
130 PRINT ::::
140 INPUT "HOW MANY RECO
    RDS WILL YOU ENTE
    R? ":B
150 PRINT ::::
160 OPEN #1:A$,RELATI
    VE,B,FIXED A
162 INPUT "DO YOU WAN
    T TO R/EAD OR W/R
    ITE RECORDS? ":B$
163 IF (B$<>"R")*(B$<
    >"W") THEN 162
164 IF B$="R" THEN 240
170 FOR C=1 TO B
180 INPUT "Name? ":A$
190 INPUT "Street, Ci
    ty/Zip? ":B$
200 INPUT "Birthday o
    r comment? ":C$
210 PRINT #1,REC C:A
    $,B$,C$
220 NEXT C
230 CLOSE #1
235 END
240 PRINT"PRESS '999' T
    O END"::
250 INPUT "WHICH RECO
    RD NUMBER? ":C
```

```
255 IF C=999 THEN 310
260 INPUT #1,REC C:A
    $,B$,C$
270 PRINT:"NAM
    E: ";A$:"ADDRES
    S: ";$:"BIRTHDA
    Y: ";C$::::
280 FOR A=1 TO 500
290 NEXT A
300 GOTO 250
310 CLOSE #1
320 END
```

Notice that the program now asks how long each record will be. This is because the file must be described as FIXED with RELATIVE. Easy to remember, perhaps, if we think that Aunt Harriet is a FIXED relative since she can never be "unaunted"! Next you see that you are to give the number of records in the file. This could be omitted to leave the file size open-ended if you wanted to.

The most important items to observe, however, are the INPUT and PRINT statement lines. Here we specify the exact record number in the file with the Basic word "REC". This is a first level "pointer" like we discussed above. It is the key to where the record is and where the computer is to look for the requested data. Here you can see quite readily that "random" really takes on the meaning of "able to move about freely in an *orderly* fashion" but is a very deliberate directed jump to an individual record in a particular file. The program listed here is of little practical value. It is not intended to be. It is offered as an illustration of data file structures and as a toy for you to play with, to elaborate upon and develop into a useful program of your own fashioning. As it stands in present form it is but one step removed from the ylem. I suggest that you do just that — play with it and see how you can embellish it and form it into something more useful. Afer a bit you will see how clear and relatively easy it is to understand the workings of file structures.

Some closing comments. RESTORE that is used with DATA statements can also be used with data files. One can "RESTORE #1" which means "make the next record you read the one at the beginning of the file" or "RESTORE #1, REC 5" which means "make the fifth record in the file the next one." Remember that with disk files you have the luxury of naming each file and that if you ever save a file with the same name as one already on the disk you will wipe out all previous data and replace it with the new. If such an action is done unintentionally, then there is nothing left but to weep because you will never get that first file back again. It is wise to keep a printed or written list of all of your files (which includes programs) that are on each disk. This way you can readily refer to them before creating a disaster. Too, many people are confused by Diskname versus Filename. In addition to naming your files, you can give the whole disk its own distinctive name. In this case it is possible to have a *DISKNAME* of "PROGRAM" and a *FILENAME* of "PROGRAM" with no conflict. It's the file names I am warning you about here. Since you can name a disk you can also access that disk much more specifically by using the disk name in an OPEN statement instead of a disk drive number. Hence,

```
OPEN #1:"DSK1.PROGRAMS"
OPEN #1:"DSK.PROGRAMS.P
ROGRAMS"
```

What comes after the first period is the DISKNAME and then the FILENAME appears after the second period. Opening a file in this manner insures that the computer will only try to access a disk with the name "PROGRAMS" *no matter which drive that disk is in* and nothing else. This offers much greater protection from accidental "replacement" of precious data files.

If you have programs from the I.U.G. library that use data files, I suggest that you LIST the programs or obtain a printout of them. In studying them you can see the many different ways that so many different minds have solved a particular programming problem with data files. You will find that there can be many different approaches to the fulfillment of one's special needs in a program.

## CHADS

This subsection of my column will be used from time to time to bring you items of special interest or to give you a peek into the arcana of my *grimoire* so you can learn some of the more unknown features of the TI 99/4(A).

MOTTO: NEVER LET A CHAD JUST LIE THERE, — PICK IT UP!

CALL FILES(0)?

When asked, many people have a fuzzy recollection of the command CALL FILES (#) (# = a number) but they don't know exactly what it is. A brief explanation is in order. If you have any peripherals for your system this concerns you. When you first turn on your system the very first thing your computer does is to check to see if any peripherals are connected to it. If it finds that to be true it automatically sets aside memory buffer space so that 3 different files can be OPENed *simultaneously*. For each of these files it uses 512 bytes of memory. Well, it is rather unusual for a good program to have more than 2 files open at once. An example might be to have one file open to access data on a disk and another file open to send that data to a printer. So, who gave the computer the right to steal MY precious memory from me? I did — by not telling it differently. We can change things to get back more memory if we want simply by typing in the command (no line numbers — this is NOT a program!):

```
CALL FILES(2)   (OR AN
Y NUMBER FROM 1 TO 9)
press ENTER
NEW
press ENTER
```

Note that this is a two-step command. By doing so you ask the computer to give you back the memory it took up. With this technique you will be able to load programs that before only gave you an error message. It is worthwhile to develop the habit of doing a CALL FILES(2) after power-up to have maximum memory to use.

OK, so what's this CALL FILES(0) business? If you have Memory Expansion,

Extended Basic and/or either Mini-memory or Editor/Assembler you can get even more memory for some otherwise impossible chores. Let's say that you have a program that is "TAPE ONLY, BASIC ONLY" from the catalog and you now want to transfer this to your new disk system. No way, they say because the program takes up more memory than is available when the peripherals are turned on. Even a CALL FILES(1) doesn't help. The way to transfer this to disk then runs something like this. Have all peripherals turned off at power up; load the program into memory from the tape. Then delete one half of the program, line by line, and save the half to tape. Reload the original, delete the *other* half and save it. Now turn the system off and reinstate the peripheral system. Powerup again, load the one half of the saved program from cassette, save it to disk in MERGE format. Now load the other half into memory and MERGE the saved file with the half in memory. Save the whole thing to disk again. This is a bit lengthy but worse, if you have ever tried to edit a large program through console BASIC, it takes about 20 seconds to get the cursor back after every deletion. An excruciating experience of torture!

Here's a way to eliminate the waiting. Powerup up with all peripherals on. Choose Extended BASIC and enter CALL INIT. Then enter the command CALL LOAD(—31888,63,255) and press ENTER. Then enter New. You have just made all peripherals invisible to the console and gotten back not only those 512 byte chunks but also the 900 or so bytes required for peripherals access. AND, in addition, you still have the Memory Expansion unit available to you. You can now follow the above procedure with no waiting for that cursor. When you finish you MUST press FCTN QUIT to cancel out the CALL LOAD you did. WARNING! After the CALL LOAD do not attempt to enter an OLD or SAVE DSK1.#### or your system will lock up. Too, remember to check your program first to see that it can be run through Extended BASIC because the final product of your efforts saved on disk will be in Memory Expansion format and you will not be able to load it through Extended BASIC. The usual conflicts are created by definition of graphics characters with a number higher than 143 or CALL COLOR statements that use a color set greater than 14. Other than that you should have little difficulty. And now you can have your tape version of "Fiddler on the Roof" in disk format!!

## PRODUCT ITEM

You are using Multiplan and want to do a printout of your spreadsheet. You decide it would be better if you used compressed print for 132 columns. What do you have to do? Save the Multiplan file, quit Multiplan, enter console BASIC, **write a little program to send the proper codes to the printer for compressed printing, quit BASIC, reenter Multiplan, reload your file and print it.**

Ah, my friends there is now an easier way. A fantastic new product is available to you for about $60.00. It is a chip and mini-board that installs in any Epson or TI printer and allows you to access 10 different printmodes directly from the control buttons on the printer. You can change the settings at any time without losing data in the buffer. Among other things, you can set double-strike, emphasized, italics, compressed, fine print, cancel skipover perforations and even automatic indent OR any combination of these. Works like a charm and well worth it. A fine product from a reputable company with reliable support. Hooks up in about 3 minutes from very well written instructions that come with it.

What is this wonder?

FINGERPRINT
from:
DRESSELHAUS COMPUTER PRODUCTS
P.O. Box 929
Azusa, CA 91702

If you want more information, give me a call at Library Services.

I goofed. I apologize. In Part I somehow I dropped a very important paragraph from the body of the article. It should have appeared right after the little program example in the article. Looking at it from the educational standpoint, however, I couldn't have worked out better. Because of my oversight you will have what I wanted to convey to you even more vividly impressed on your memory. Here's the missing paragraph. Remember that it goes right after the "program"!

If you have typed in this example and run it for use with a Cassette recorder, the error message will begin to impress on you some more of the restrictions that use of tape imposes on file handling techniques. Not only is tape s-l-o-w. It also makes a serious cut into the flexibility in file use that TI affords us. Look at line 160. For tape use you found out the hard way that tape does NOT allow the use of VARIABLE length files. It demands only FIXED files. By not mentioning it the file is OPENed as "UPDATE". Cassettes will also NOT allow you to use it. It must be either INPUT or OUTPUT but NOT both for simultaneous use as a disk drive would. Now perhaps you will understand why some tape based program files require the use of TWO recorders. One for *OUTPUT* and yet another for *INPUT*. A bit of a bore, eh? Waste space with FIXED lengths AND be restricted from using an UPDATE file for either reading OR writing. Think about that!

DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES

# PRICE BREAKTHROUGH

## Super Sale on New Disk Drives

Starting at **$169.00!**

**Tandon — Siemens — Remex — MPI — Teac — Shugart — Tabor**

40 or 80 Tracks — Single or Dual Head — New 3½" Drivette™

Our Disk Drives Are Capable Of Single **And** Dual Density Operation

The **NEWEST** Technology Capable Of Operating On Most Popular Computers

**Drive a Hard Bargain!!™ — 5 M.B.-20 M.B. Complete Systems . . . . . from $999.95**

**Diskette Breakthrough — 10 Pack in Library Case — $~~18.95~~ NEW LOW PRICE**

**Since We Are Always Finding Ways To Save You Money, Please CALL For Our Most Current Pricing.**

**TOLL FREE ORDERING**
**1-800-343-8841**

**GENERAL AND TECHNICAL**
**1-617-872-9090**

| | |
|---|---|
| Model I/III/IV Drives (0 1 2 3) | starting at **$169.00** |
| **Color** Computer Drive (0 1 2 3) | **$ Call Toll Free** |
| Apple/Franklin Compatible Add-On Drives with Case & Cable | ~~$219.95~~ NEW LOW PRICE |
| Apple/Franklin Compatible Drives with Controller (Now 40 Tracks) | **$259.95** NEW LOW PRICE |
| Model I/III/IV Memory Upgrade | **Call Toll Free** |
| **Printers** — Daisywheel/Dot Matrix | **$ Call Toll Free** |
| Diskettes in Library Cases | ~~$18.95~~ **New Low Price** |
| **Cases and Power Supplies — (Single-Dual-1/2 Height)** | starting at $44.95 |
| Printer **Buffers** 8K to 512K | starting at $143.95 |
| Percom Double Density Controller (Model I) | **$ Call Toll Free** |
| Holmes Model I/III Speed-up Mod-VID/80 | starting at $90.00 |
| Color Computer Printer Interfaces | starting at $29.95 |
| Cables — Printer/Disk Drive | starting at $23.00 |
| DOSPLUS | **$ Special Prices** |
| **Repair Services** Now Offered — FAST Turn-a-Round | **$ Call Toll Free** |

**Warranty on Disk Drives — 6 Months — Extended Warranty $ Call Toll Free**

# SOFTWARE SUPPORT, INC.

One Edgell Road, Framingham, MA 01701 (617) 872-9090

Hours: Mon. thru Fri. 9:30 am to 5:30 (E.S.T.) Sat. 10 am to 4:30 pm

**DEALER INQUIRIES INVITED.**

TERMS:
M.C./Visa/Amex and personal checks accepted at no extra charge.
C.O.D., please add $3.00.
Shipping: Please call for amount.
Not responsible for typographical errors.



**CANADA**
**MICRO** R.G.S. INC.
751, CARRE VICTORIA, SUITE 403
MONTREAL, QUEBEC, CANADA, H2Y 2J3
Regular **Tel. (514) 845-1534**
**Canadian Toll Free 800-361-5155**

**Service! Service!**

*All in stock products are shipped within 24 hours of order. Repair/Warranty service is performed within 24 hours of receipt unless otherwise noted. We accept C.O.D., foreign and APO orders. Schools and D&B corporate P.O.s accepted.*

DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES

# SOFTWARE

# THIRD PARTY PRODUCT REVIEW

**By Dana Nichols**
Managing Editor

When the news that Texas Instruments had withdrawn from the Home Computer market reached our ears on October 31, 1983, it appeared that everyone else was expected to follow suit. Not so.

Third party sources have been producing software and hardware lines for the TI 99/4 and 99/4A since the Home Computer's introduction in 1979. They realized then that they were little fish in a big pond, and worked their way to the surface by producing software equally as good, if not better, than the contributions TI made to their own computer line.

The International 99/4 Users-Group has conducted an extensive overall evaluation of the third party market, and are more encouraged than ever that both hardware and software producers will continue to bring quality products to owners of the 99/4 and 99/4A. Listed below are products researched and evaluated by the IUG; we hope you find them to be equally enjoyable and useful.

**TITLE: NAMES**
**AUTHOR: THinc.**
**PURPOSE: Education**
**LANGUAGE: BASIC**

This colorful game for children ages two to six helps them recognize names associated with objects while providing a colorful background and a pleasing melody.

Your child will see a meadow with green grass and a clear blue sky, in which people and objects will appear after he or she spells the name. For example, at the bottom of the screen, the letter 'G' will flash. Your child presses the 'G' key, and a short melody will play to signify your child's correct response. The next letter, 'I', will then appear, as will subsequent letters to spell 'GIRL.' Should your child press a wrong key, a two note melody will play that sounds much like "uh oh." Once the entire name has been spelled, a light melody will play as the girl appears in the meadow.

The word 'GIRL' will appear next to the figure of the girl on the screen throughout the game/lesson. This will help your child to not only associate the word with the girl, but will serve to help your child remember the way the word is spelled.

There are many objects for your child to learn to recognize and spell, and they will love the music! A thoughful option includes the ability to personalize the program to use family names.

NAMES is one of many console BASIC educational games for your child aged two to six. More sophisticated programs in Extended BASIC are available not only for your young children but parents and teachers as well. Console BASIC versions, $14.95, and Extended BASIC versions $19.95, from THinc., P.O. Box 6129, Ft. Wayne, IN 46806, will provide your children with hours of fun as they learn the fundamentals of spelling and recognition.

**TITLE: Diablo**
**AUTHOR: Extended Software Company**
**PURPOSE: Entertainment**
**LANGUAGE: Extended BASIC**

Depending upon how well you plan your moves, Diablo, from Extended Software Company, could keep you at the console for hours! Derived from the European game, DIABLOTIN, your screen is filled with 232 sections of track, two tracks on each of 116 movable panels. The purpose of the game is to keep the ball moving on a continual course. It's much harder than it sounds!

You have one blank section with which to work. When you move your section UP, the section you replace moves DOWN, and so on. Each section has a different pattern, and moving the sections creates a new pattern each time. You must plan ahead and position your track to enable the ball to continue to roll. The game involves strategy and intense concentration for serious players.

Each time a section is traversed by the ball, the section disappears, thereby eliminating track footage. Should your ball roll off the end of the screen, or travel to the end of a section with no continuous path in sight, the ball will roll down to the bottom of the screen and the game will be over. Once 60 sections have been eliminated, the track turns green and a wrap-around screen may be utilized. When the ball goes off one end of the screen it will appear at the opposite border, provided there is a compatible track to receive it. You only get one chance per screen, so stay on your toes!

Simplistic strategic challenge and excellent graphic resolution make this game truly worth its $19.95 price, now available from Extended Software Company, 11987 Cedarcreek Drive, Cincinnati, OH 45240 (513) 825-6645.

**TITLE: Companion**
**AUTHOR: Intelpro**
**PURPOSE: Word Processing**
**LANGUAGE: Assembly**

An extremely fast word processor written in Assembly Language, Companion requires the following: a. Version 110-Extended BASIC; b. Satisfactory use of the available 30 columns and all 24 available rows on the television or monitor; c. Disk controller card and one disk drive; d. 32K Memory Expansion Card.

The manual included with Companion is well-written and easy to understand. In addition to the four main features incorporated into Companion, namely Editing and Printing, Saving and Loading Texts, Batch Processing, and Parameter Revision, Companion now offers String Search, Text block manipulation, Single sheet operation and Users customization.

This versatile word processor reformats the screen up to 30 times per second. There are 11 control characters which are used for paragraph indentation, line feed, centering text on a lines, and several other useful operations. In addition, if an error is made, Companion directs you to the exact program line number where something went wrong, enabling you to devote more time to editing than correcting.

We at the IUG feel that Companion is the finest word processing program on the market today. It is available on diskette for $79.95 from Intelpro, 5825 Baillargeon St., Brossard, Quebec, Canada J4Z 1T1 or through the IUG.

**TITLE: Treasure Hunt II**
**AUTHOR: Mirage Software**
**PURPOSE: Entertainment**
**LANGUAGE: Extended BASIC**

Treasure, treasure, everywhere! Mirage Software offers you hours of challenge with Treasure Hunt. A four-screen graphic adventure, Treasure Hunt features an unplundered tomb loaded with treasure. Your mission is to get all the treasure in the first room, maneuver through the second and third, and in the fourth room, find the treasure that holds the key to the secret exit.

The first room features a four-floor area with slides connecting the floors. You must pick up all the treasure to enable you to get to the next room. Sound easy? Not so! No treasure-filled tomb

goes unguarded, and deadly creatures try to impede your progress on the ground while ominous spiders hover overhead. You must jump over the creatures to gain your treasure, but watch out for those hanging spiders!

Blocks and slides serve as the second room. You must jump from block to block, gathering treasure as you go. Again, creatures are ascending up the blocks and your jumps must be timed perfectly (especially at the bottom!) Consisting of four floors, you can reach the next lower floor by sliding down the slide. You cannot, however, go back up so you must be confident in your step!

The third screen features a rope chamber and blocks you must reach while again avoiding those deadly monsters and spiders. This time, the spiders use the ropes to climb while the monsters have gained a weightless advantage. You must jump diagonally down to each block while avoiding the upward moving spiders and monsters. Again your timing must be precise.

The final screen holds the treasure that contains the key to the secret exit. You must travel across the four floors and down the three ladders while avoiding airborne and ground-moving obstacles. Again, pick up all the treasure you can and hope that one of them holds the key to freedom! Completion of the fourth screen signals the first screen to return at a more difficult level, and you will gain an extra life as a result of your success!

Treasure Hunt II offers hours of entertainment and challenge. Available for $19.95 on cassette or disk, Treasure Hunt requires Extended BASIC. In addition, you have your choice of joystick or keyboard control. Contact Mirage Software at P.O. Box 613, West Seneca, NY, 14224, or call (716) 674-2203.

**TITLE: Oscar**
**AUTHOR: Databar Corporation**
**PURPOSE: Bar Code Scanning**

OSCAR, a revolutionary new way to enter data into your computer, is now available from Databar Corporation, Minneapolis, MN and the IUG.

OSCAR (Optional Scanning Reader) employs optical bar code scanning, most recently used in the form of Universal Product Codes of grocery store products. Software is printed directly onto paper in bar code form, and grooves in the template guide the tip of OSCAR's "wand," a palm-sized box attached via a cord to OSCAR. Simply slide the tip across one bar code line, left to right; move the tip down to the next groove, and slide the wand back across, right to left. The scanning system in the device "reads" narrow and wide black bars separated by white space and picks up the digitally-encoded information in the bars in machine-readable form.

At present, the only software used with OSCAR is available from Databar Corporation. Magazines published by Databar will contain programs in bar code form which will be about four pages long and require two to three minutes' scanning time. Each magazine will contain eight programs in addition to articles and other useful information, and is priced at $10, or $1.25 per program.

OSCAR operates using four "D" batteries, is activated when the wand is removed from the cradle and shuts off when the wand is returned. A "time-out" feature shuts power off when the wand is not returned or is left unused for 60 seconds.

Information concerning OSCAR can be obtained by contacting Databar Corporation, Customer Service Department, 10202 Crosstown Circle, Eden Prairie, MN 55344, or the IUG.

**TITLE: The Learning Center**
**AUTHOR: Intellestar**
**PURPOSE: Education**
**LANGUAGE: Extended BASIC**

The Learning Center package serves to provide teachers and parents with a set of four games, all of which incorporate a file of questions for a child to answer.

The adult composes a question file by using the File Editor included in the package. This enables the parent or teacher to compose files according to the capabilities and age level of the child.

The Learning Center package includes the following:

TLC File Editor
Sample File
Space Patrol — Lost!
T.V. Sweepstakes
Last Jellybeans on Earth
Baseball

The player chooses the game he wishes to play and a question file is chosen. Both files are loaded and the main menu screen will appear. Until the child is comfortable with the package, I would advise adult supervision at this point. While the game programs are involving and fun, the directions are somewhat more sophisticated.

Although the question files may be composed to test any level of knowledge, the games into which they are incorporated are directed at younger children, in the 6-12 years old range. The graphics are good in each game; however, children such as advanced junior high and older students may tire of them.

Baseball and T.V. Sweepstakes are two-player games, while Space Patrol — Lost! and Last Jellybeans on Earth are one-player contests. By using the TLC File Editor, you as an adult can create question files in areas in which your child may be experiencing difficulty.

Extensive documentation is included with the package. For more information concerning The Learning Center, contact Intellestar at 25 West Middle Lane, Rockville, MD 20850.

*(printout on page 51)*

# PROBE A SOFTWARE REVIEW

**By Dana Nichols**
Managing Editor

It's general knowledge that beginning programming can be a headache to many new computer owners. Young and old alike search for material to help them understand programming concepts and write successful programs.

Scott, Foresman and Company have developed Probe, a curriculum for children in grades 1-8 wherein they can begin to understand the basics of computer programming. It is presently available for use with cassette or disk. Using Probe, you as an adult can help your child better understand programming with the 99/4A Home Computer.

Probe was written in 1982 by graduate seminar participants who subsequently served as instructors. The curriculum concepts and computer programs included in the workbooks were field tested during the 1982 Wisconsin Summer Computer Fest.

If saving programs on cassette, all that is needed is a cassette recorder and blank cassettes. If you own a disk drive, you will need the Disk Manager module in order to initialize your disks.

The workbooks are offered at three grade/age levels: primary, grades 1-3; intermediate, grades 4-6; and junior high, grades 7-8. Probe combines workbook activities with hands-on experience and training in a motivating format.

Each Probe package contains the following:

Student workbook with motivating activities and concise instructions.

Instructor Edition, exactly like the workbook, with additional margin notes and overprinted answers.

Diskette or cassette, enabling you to sample, display, or reference segments of the longer programs in the workbook.

Wall charts concerning the Keyboard and a Command chart, containing BASIC glossary terms.

Pad of screen grids for plotting.

The workbook is designed for use in an instructional format, and total supervision is needed for younger children. Once the children in grades 1-3 learn the alphabet and basic concepts of numeration, they will be on their way to easy programming with Probe.

Probe is not designed to teach your children the fundamentals of reading and math. It is designed to help your child learn to program; to enable him to issue commands and see them carried out. Adults must keep this in mind when they first open the Probe Instructional Manual for their children ages 6-8; it is not a "SEE JANE RUN' type workbook. It is more along the line of 'PRINT "SEE JANE RUN"

The wall charts are informative and explanatory. Again, younger children will not be able to immediately read the material, but repetition and recognition will have your child breezing through Probe in no time at all. Chances are, learning with Probe will make his general reading classes easier to master.

The keyboard chart highlights the most often used keys, such as Alpha Lock, ENTER and FCTN, and explains their functions. In addition, Special Reminders are included on the chart such as "You cannot substitute the letter I for the numeral 1."

The Computer Commands chart deals with typewritten commands such as CALL CLEAR, GOTO and SAVE. Following each COMMAND is its Meaning and How it is Used. For example, COMMAND: CALL SCREEN; What it means: Changes the color of the computer screen; How it is used: 20 CALL SCREEN (13).

In addition, the Computer Commands are found at the end of the workbook, as is a short glossary of Computer Terms such as Bug (a mistake in a program), Horizontal Line (a line that goes across the screen), and Software (programs written for the computer).

The following are highlights of the Workbook designed for grades 1-3, Beginning BASIC Programming Activities.

The Table of Contents is composed of Writing Your Name, Sentences and Numbers; Saving and Loading Programs on Diskette and Cassette; Drawing Color Pictures on the Computer; Learning More About Drawing in Color, Using Words and Number as Strings, Learning About Bugs and Debugging; Teaching the Computer to Count; and Adding Sound to Your Programs. Each chapter incorporates the previous lesson to provide your child with a smoothly-running, orderly presentation.

The lessons also incorporate terms your child will immediately understand and begin to recognize; terms such as school, home, park, etc. will help lessen any fears your young child may have in tackling a computer.

The first chapter, for example, shows your child how to print his or her name and a friend's name. The directions are as follows:

'To tell the computer your name, type this line. Press the ENTER key after you type each line. To make the quotation marks, hold down the FCTN key and press the P key.'

PRINT "your name" ENTER

Now type this line.

PRINT "a friends name" ENTER

```
TI BASIC READY
   PRINT "LISA"
   LISA
   PRINT "BEN"
   BEN
```

Type the words CALL CLEAR and press ENTER. What does CALL CLEAR do?

Probe uses terms your child will feel comfortable with. For example, he is then instructed to 'Use PRINT to write each of these words or sentences on the computer.'

| | |
|---|---|
| HELLO | SUMMERTIME |
| DAD | I LIKE GAMES. |
| MY CAT | WOW, THIS IS FUN. |

Your child is then encouraged to use his imagination. His next instructions are to 'Write some words or sentences below.' This will give you as a parent or teacher an idea of his learning capacity by observing how he handles his first solo command.

Probe offers much to your children in grades 1-8. It helps them learn the basics of computer programming, teaches them the principles of discipline and accuracy, and helps prepare them for the high technological world ahead. Probe is available now from Scott, Foresman and Company, Glenview, IL, and will soon be available through the IUG.

# PROGRAMMING "TIPS" FROM AMLIST A BOOK REVIEW

**By Terry Heim**
*Staff Technical Editor*

With the ridiculously low price of the 99/4A Home Computer, many people have purchased it with an "it's finally cheap enough, let's get one and see what it can do" attitude. And with the appropriate software, the computer has proven to be invaluable for education, household finances, number crunching of all kinds, and home entertainment. But for those of you who would like to learn more about how to seriously program the 99/4A and get the most out of your investment, you may be finding out that you need a few TIPS.

The BASIC TIPS INSTRUCTIONAL Manual from AMLIST, Inc. teaches programming from the first concept of a program to advanced topics in data file management and memory conservation. TIPS began in January, 1983 as a series of 12 lessons that were mailed out once each month. Now these lessons have been combined into one book. Although TIPS proves invaluable to the individual, schools and business concerns will also find the TIPS program suited to their needs.

TIPS is designed to teach more effective use of the commands and functions of TI BASIC, not merely their definitions. In other words, TIPS is not a revised and reworked Users Reference Guide packed with each console; it is a programming course that is built around TI BASIC and the 99/4A console. The manual includes many short examples and 16 major programs ready to type in and the "hands on" experience gained by actually inputting the programs is very important.

TIPS is divided into 12 chapters. Chapter one contains instructions and manual review. It serves not only as an overview of the reminder of the book but as a preliminary "pep session," encouraging novice programmers to read carefully, enter the examples in the manual, and most importantly, not to become discouraged when they cannot program arcade-style games or professional level business applications within three hours of picking up the manual.

Chapter two discusses programming philosophy and introduces console BASIC. It is in this chapter that you will perhaps type in your first program.

Chapter three discusses error messages and instructs the user in the art of using these messages as a programming tool instead of something to be dreaded or feared. This chapter also emphasizes the importance and methods involved in debugging a program. Testing procedures are pointed out and some "tips" on how to find those hard to track down logic errors are also included.

Very user-friendly, you'll never feel alone when learning to program with TIPS. The following is an excerpt from the text in chapter three concerning Debugging and Error Messages: 'We're going to use the "KAMAKAZE RUN" program as an example again and give you the general sequence of events which took place in the writing of this program. This may seem to be a digression from the "debugging" topic,but it's our philosophy that half the battle is knowing where an error occurred. By keeping each step small and running the program frequently we keep the possibilities to a minimum. As you can imagine, you'll feel better jumping into programming with this step-by-step method of debugging.

Chapters four and five consist of information on graphics and music, respectively. A table of the character codes needed to display a magnified alphabet is included along with many musical subroutines.

Chapters six and seven deal with by far the most complicated and confusing aspect of computer programming: manipulation of data. Chapter six deals with data files while chapter seven discusses arrays. Special emphasis is placed on the most economical ways to store data on tape; however, the principles used to save data can easily be applied to disk drive operation. Although a thorough presentation of data handling would require a book many times the size of TIPS, this manual sets forth a foolproof, easy to understand and memory saving method of organizing data.

The latter part of the text deals with more advanced topics such as sorting, testing and condensing programs. Although TIPS was written toward the beginner, even the experienced programmer should be able to benefit from these chapters and increase his programming proficiency.

## THE PROGRAMS

TIPS makes the instructional material practical by including many examples and subroutines and 16 major programs. The programs are printed in "screen-compatible" format, meaning that programs are printed on the page in 28-character wide columns so that the programs appear on the screen exactly as they are on the page. There are one or two programs in each chapter ranging in scope from "Monkey Business" to "Budget Maintenance."

Each of the programs was created to highlight a specific technique or idea and all are thoroughly documented, either in the program itself or as part of the instructional material surrounding it. Six of the programs use data files and a couple of them push the capabilities of the unexpanded 99/4A computer to the limit.

In particular, I was impressed with the "Budget Maintenance" program. This program does the same job as several of the programs in the Owner Written-/Translated Catalog. The program provides for the creation of a chart of accounts (including the account name, a monthly budget figure, and a year-to-date total of amount charged to that account for each account on the chart) and for the entering, retrieval and revision of up to 71 checkbook entries. The program then saves all of this information on tape for later use.

Keep in mind that all of the programs in TIPS are written to run using no extra peripherals; the only equipment you need is a program recorder.

An another example, the "Gold Handicap" program stores and determines

handicaps on a base of 20 rounds of golf for up to twelve players. In addition, eight individual rounds may be entered for hole by hole calculations and comparisons.

These are two of the more complicated programs; however, the point is, you don't have to be a genius or a mathematician to write complex and useful programs.

Chapter 12 concerns the Summary and Looking Ahead. It is a basic review of the instructional manual and points out major topics where you should, by now, be well versed. One valuable point the author makes is this: "When you begin to think about every potential computer application in this manner (breaking the program down into parts) you'll be well on your way to being an expert programmer...(and) there will be very few tasks that you can't complete."

The author also points out areas in which you can plan to expand your computer, such as Extended BASIC, Speech Synthesizer, Printer/Peripheral Expansion, Disk Drive, Modems, Additional Memory, and Third Party Peripherals. A short explanation of what each field would require is included as well as advantages and disadvantages of each.

## CONCLUSION

TIPS is directed at the semi-serious novice who wishes to take the time and put forth the effort to learn computer programming. TIPS is not an "instant programmer" pill. The overall tone of the text is not particularly humorous or exciting but is filled with quality information useful to the amateur and experienced programmer alike.

The programs, in addition to demonstrating some of the techniques discussed in the manual, will prove to be a strong base on which to develop a software library. TIPS is available from AMLIST, Inc., 4542 Memorial Drive, #202, Atlanta, GA 30032, at phone #1-800-241-6083. If you are looking for an inexpensive, easy to understand way to learn to program, TIPS is for you.

# NOW WHERE WAS THAT LINE?
# FIND IT WITH STRUCTURED PROGRAMMING

**By John Phillips**

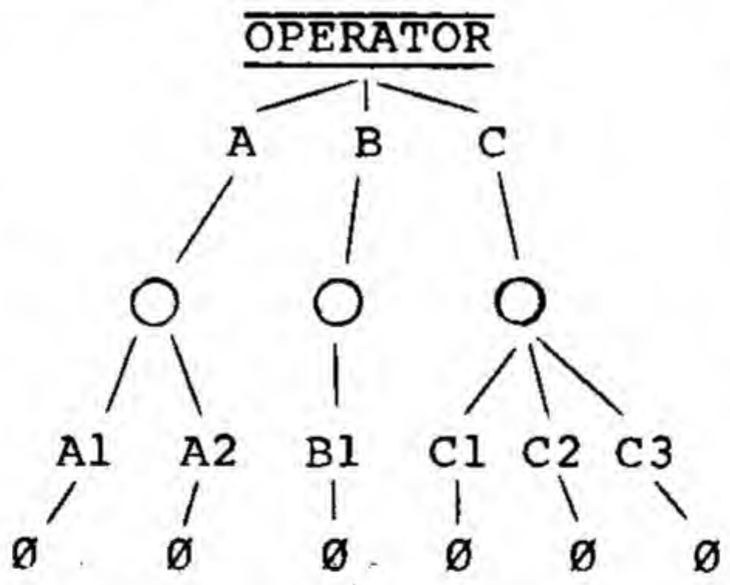
My topic in this piece of journalism describes a method that can rid you of the frustrating "debugging" time that so many precious hours are spent on. This method is called STRUCTURED PROGRAMMING.

I am quite sure that most computer programmers have had the term "structured programming" beaten into their brain. What I have discovered is that most of the time this technique is either ignored or forgotten since it adds time to the development phase of a project. Let me assure you this: structured programming ALWAYS pays off in the long run.

A former college professor of mine made an unusual comment about a COBOL program that I had written, which was an unbelievable mess of code. He fondly called it "spaghetti code." There was absolutely no structure, no use of subroutines, no comments, and it made no SENSE. It turned into a nightmare when the instructor handed out the next assignment . . . modify the previous program to perform some other repetitious task. All I could think of was "how in the world am I going to modify this program to do that?" Structured programming was the answer to my problem.

Needless to say, I rewrote the entire program from a TOP-DOWN approach (more on this later) and all further enhancements to that program (of which there were quite a few) were incredibly simple to implement. Let me state that structured programming lends itself well to ALL programming languages, not just COBOL. I could not even begin to imagine an Assembly language program without structure. Let's find out what structured programming is all about.

Imagine that you are a conveyor belt operator whose responsibility is to send parts down a set of conveyor belts and receive modified parts from those same conveyor belts. In addition, each separate conveyor belt can branch off into one or smaller conveyor belts, if needed. Examine this picture:




In this representation, the operator is at the head of the conveyor belt process. The three main belts are marked A, B, and C. Notice that belt A can branch off into either A1 or A2. The same type of situation applies for belt B and belt C. Let's suppose that you have a part that needs to be built into some useful product. The process flow is as follows:

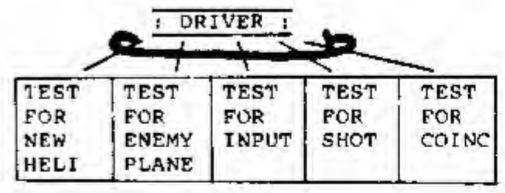
1) Send a part down belt A. Receive a modified part back.
2) Send the modified part down belt B and receive back a part with further enhancements.
3) Send the enhanced part down belt C and receive back the finished product, ready to ship.

As the operator, you are not concerned with what happens in the lower belts. Your only concern is that you get something back from belt A and pass it on to belt B and then to belt C. You will repeat that process until all parts have been used or some other condition stops the process. From the operator's position, you view the system as a TOP-DOWN standpoint. In other words, the main flow of action is defined by belts A, B, and C. The lower, or SUBORDINATE belts were defined after the superior belts. The operator knows that once he passes a part down belt A he is going to receive back a modified part through the action of either belt A1 or belt A2 or both. The main point is that the operator is not concerned (at this point) with what happens in the major flow of the superior belts. Once you define the major flows of the program you are designing, you have developed the DRIVER of the program. It will steer the direction of the program . . . in essence, it is the "operator" of your program flow.

In "Helicopter Attack" (published Enthusiast '99, Vol. 1, No. 1, May 1983) I defined 5 "superior belts":

1) Determine if a new helicopter is needed.
2) Determine if an enemy plane is needed.
3) See if the player wants to move.
4) See if the player wants to fire.
5) Determine if the player's shot has struck any object.

At this point, a process chart can be drawn.




Notice how the driver is an infinite loop. The driver is not concerned with how many helicopters have been hit, or how many shots have been fired. Its only

(continued on page 28)

Norton Software
P.O. Box 3574 South
Halifax, N.S. B3J 3J3

**TI99/4(A)**
**Commodore 64**

# 6 Reasons Why We're the "Name of the Game"™

1. **Cross Country Car Rally.**
   Fast action road race.
   TI99/4(A) (Ext. Basic only) (K) ..... **19.95**
   Commodore 64 (Machine Code) (K) ... **19.95**
   Cassette

2. **Attack-man.** More than just a maze game. (Ext. Basic only) (K/J)
   TI99/4(A) ..... **19.95**
   Cassette

3. **Super Frogger.** Cross 5 lanes of traffic, 2 barges, 3 logs, treacherous pirhana infested waters and get eaten by the alligators anyway!
   TI99/4(A) (Ext. Basic only) (K/J) ..... **19.95**
   Cassette

4. **3-D Startrek.** Adds an exciting new dimension to Startrek.
   TI99/4(A) (Specify Ext. or console) (K)
   ..... **14.95**
   Cassette

5. **Lunar lander.** Crash on your favorite planet; many levels.
   TI99/4(A) (Specify Ext. or console) (K) . **14.95**
   Cassette

6. **Tank.** Blast the computer or an opponent. TI99/4(A) (Specify Ext. or console) (K/J) ..... **14.95**
   Cassette

Diskette versions are $4.00 extra and require use of the optional memory expansion unit.

Add $2.00 shipping and handling charges for less than three games.
(K/J) Keyboard and Joystick versions included
(K) Keyboard only

**Receive a 20% discount if you order 3 or more games.**

All programs available in Ext. Basic unless otherwise indicated in the description. Please Specify.

MasterCard VISA

Look for your Norton Software games in your nearest store, or call 1-902-425-8300 (orders only please) or write for our free catalogue showing these and other games.

**Norton Software will continue to support the TI99/4(A) Home Computer for many years to come. Dealer Inquiries Welcome.**

# NOW YOUR COMPUTER FITS THE ORIGINAL ARCADE HITS.


Now the excitement of original arcade graphics and sound effects comes home to your computer.

Introducing ATARISOFT.™ A new source for computer software.

If you own a Commodore VIC 20 or 64, a Texas Instruments 99/4A, an IBM or an Apple II, you can play the original arcade hits.

DONKEY KONG by Nintendo, CENTIPEDE,™ PAC-MAN, DEFENDER, ROBOTRON: 2084, STARGATE and DIG DUG. (On the TI 99/4A you can also play Protector II, Shamus, Picnic Paranoia and Super Storm.)

So, start playing the original hits on your computer.

Only from ATARISOFT.

Some games also available on ColecoVision and Intellivision.

**ATARISOFT™**

Now your computer fits the arcade hits.



Complete this coupon and we'll keep you up to date on the newest hits from ATARISOFT.™

Name _____

Address _____

City _____ State _____ Zip _____

Telephone _____

PRODUCT OWNED: (Check one)

1 ☐ TI-99/4A
2 ☐ IBM PC
3 ☐ Commodore 64
4 ☐ ColecoVision
5 ☐ Commodore Vic 20
6 ☐ Intellivision
7 ☐ Apple II
8 ☐ _____

Mail to:
Atari, Inc., P.O. Box 2943,
So. San Francisco, CA 94080.

ASM 22

(continued from page 24)
concern is to make sure that the designated processes are carried out. The driver gives and receives information to and from the subordinate modules. That is the ONLY way the subordinate belts receive information . . . it is given to them by the driver. There is no belt that connects belt A and belt B. All data must touch the hands of the operator before any other major belt can modify it.

If you can grasp everything I've just talked about, you now have an idea of what structured programming is all about. Design the package from the TOP DOWN. Define the major tasks of the program first, and worry how they are to be accomplished later.

Once you define the tasks (belts) needed, you can write the driver without fear. Here is how I would write the driver for the above process chart. I will display them in BASIC, Extended BASIC, and Assembly Language:

BASIC

```
10000 GOSUB 1000
10001 REM TEST FOR NEW HELI
10020 GOSUB 2000
10021 REM TEST FOR ENEMY PLANE
10030 GOSUB 3000
10031 REM TEST FOR PLAYER INPUT
10040 GOSUB 4000
10041 REM TEST FOR COINCIDENCE
10050 GOTO 10000
10051 REM KEEP DRIVER GOING
```

EXTENDED BASIC

```
10000 CALL NEW_HELI          !TEST FOR NEW HELI
10010 CALL ENEMY             !TEST FOR ENEMY PLANE
10020 CALL BASE_MOVE         !TEST FOR PLAYER INPUT
10030 CALL SHOT_CHECK        !TEST FOR PLAYER SHOT
10040 CALL SHOT              !TEST FOR COINCIDENCE
10050 GOTO 10000             !KEEP DRIVER GOING
```

ASSEMBLY LANGUAGE

```
MAIN   BL  @NWHELI     !TEST FOR NEW HELI
       BL  @ENEMY      !TEST FOR ENEMY PLANE
       BL  @BASEMV     !TEST FOR PLAYER INPUT
       BL  @SHTCHK     !TEST FOR PLAYER SHOT
       BL  @SHOT       !TEST FOR PLAYER COINCIDENCE
       JMP MAIN        !KEEP DRIVER GOING
```

At this point, I feel very confident that I have written a 100%, error-free piece of code. From this point, I would start defining the lower belts for each of these main belts, one by one. Once again, follow the TOP DOWN approach to define what subordinate modules you will need for each main module. When you can no longer define modules to perform the necessary tasks, you have reached the bottom-line of coding and you should be looking at a clearly written piece of code. I would suggest you study "HELICOPTER ATTACK" to get you started on your quest for structured programming techniques.

I had mentioned earlier in this article that the use of structured programming always pays off in the long run. Well, it does. Whenever I encounter a bug (a "logical" bug, not a syntactical bug) I immediately know in which "belt" or subroutine the problem is located. This saves hours of program tracing to isolate the culprit, since I know exactly where the bug originates from. There are some rules that I would like to pass along to you to help make structured programming even more efficient. Here are some rules of thumb:

1. Isolate the major functions of the program to be designed.
2. For each function, develop a subroutine to handle that function or to call other subordinate modules to accomplish that function.

3. A single subroutine should be no more than 50 program lines long. If it exceeds 50 lines, chances are the subroutine can be broken into two or more subroutines.
4. Make the driver on a large line number . . . say 30000.
5. Make all major function subroutines on even, repetitious line numbers, i.e. 5000, 10000, 15000, etc.
6. Make all subordinate subroutine line numbers subsets of the superior calling subroutine. For example, if the superior subroutine starts at line 1000, make all subordinate subroutines at 1100, 1200, etc.
7. Use meaningful variable names, such as HELI-COUNT instead of HC or H.
8. EVERY subroutine must have a minimum 3-line comment block surrounded by noticeable characters to abbreviate the function of the subroutine.
9. Comment every line of the driver.
10. EVERY subroutine shall have one entry point into the subroutine and one exit point from the subroutine . . . to prevent "spaghetti code."
11. At the beginning of the program, display a large comment block with the following information:
    A. TITLE
    B. AUTHOR
    C. AUTHOR'S ADDRESS
    D. DATE STARTED AND FINISHED
    E. VERSION NUMBER (1.0 INITIALLY**)

    **After every modification to the program, update the version number (i.e. 1.1) and add a comment describing the condition requiring the modification, the date fixed, the line numbers modified (or added/deleted), and how the problem was resolved.

I hope this article has shed some light to those of you who grapple with your own code, trying to figure out just what you had done six months ago or are trying to decide the best way to code a program. Structured code is easy to read, easy to modify, and makes finding errors a whole lot easier. Personally, I would not begin to examine a program that was not structured . . . I would quickly lose my sanity. Whether it be BASIC, Assembly, PL/I or COBOL, anyone can read my code and get a general idea of what is going on. I hope all the readers of this article can make that same claim . . . if not, keep on structuring!

# FIND OUT WHY TENEX IS #1 . . .

Mail or call toll-free today for your FREE new "Everything Book for TI Home Computers". Learn everything you can do with your computer, plus hundreds of software, hardware, and accessory items . . . all for the 99/4A, and all at super discount prices. Shy of mail order? Try your friends at TENEX, then compare service and value!

## $2.95 DUST COVER SUPER SPECIAL

Why for so little? Just so you will get to know us! The best quality (honest) heavy gauge, antistatic translucent vinyl. Sewn seams to fit your 99/4A console. Will accommodate cables and speech synthesizer. Attractive protection for your investment. #10017

*(Limit One)*

## Serious About Games?

only $21.95

Plug directly into console for arcade style action with these famous WICO® joysticks. Requires TI adapter.

**Wico Joystick** — Comes with fire buttons on handle and body, 5' cable. Retail $29.95. #10252 . . . . . Only **$21.95**

**Famous Red Ball** — The kind you find in arcades. Fire buttons on handle and body, 5' cable. Retail $26.95. #10266 . . Only **$23.95**

**NEW. "The Boss"** — An incredible price that includes pistol-grip controller, fire button on handle, 5' cord. #19707 . . . Only **$15.95**

**TI Adapter** — For up to two standard joysticks. #10285 . . . . . **$11.95**

## The "Everything Book for TI Home Computers"

### EVERYTHING YOU NEED

Printers, joysticks, books, disk drives, interfaces, dust covers, program recorders, CABLES, disks and tapes, surge protectors, x-y plotters, MODEMS, Monitors, supplies, and more!

All first quality from major manufacturers with the best prices, and all in the FREE "Everything Book for TI Home Computers."

*(A $2.00 Value)*

### UNEQUALLED SOFTWARE SELECTION

Over 600 titles for education, games, finance, business, graphics, and much more. On disk or cassette from these leading software publishers...

American, Dynamic Data, EduCAItor, Easyware, Extended, Funware, Futura, Gembar Graphics, Harvey, Headwind, Instant, Lecher Offshore, Linear Esthetics, Moonbeam, Microcomputers, Meca, Micro-Ed, Maple Leaf, Micro-Pal, Not-Polyoptics, Norton, Oak Tree Systems, Pewterware, Professional Microware, Program Design, Republic, Romox, Simulsoft, Tomputer, Scholastic Wizware and more . . .

All in your FREE "Everything Book for TI Home Computers"

### ALPHACOM PRINTERS

- No Peripheral Expansion Box required!
- Silent thermal printers
- 80 characters per second
- High resolution printhead

$109

**Alphacom 42** is a 40 column wide printer
#20529 . . . . . **$109.00**

**Interface 42** cable for 99/4A console . . . Call

**Thermal Paper** (42)
#20567 (2 rolls) . . . . . **$4.75**

**Alphacom 81** is a full 80 column printer
#20372 . . . . . **$149.00**

**Interface 82** cable for 99/4A console . . . Call

**Thermal Paper** (81)
#20619 (2 rolls) . . . . . **$9.50**

**TENEX Computer Marketing Systems**
**Box 6578, South Bend, IN 46660**
**Please call 219-277-7726 with questions.**

# ORDER TOLL FREE 800-348-2778

MAIL ME TODAY

(use this form or copy your own)

☐ Please send me the FREE "Everything Book for TI Home Computers." NO PURCHASE NECESSARY.

Name _____

Address _____

City/St/Zip _____

Phone _____

*Thank You!*

**Mail to: TENEX, Box 6578, South Bend, IN 46660**

Pay by Check, ☐ Visa, or ☐ MasterCard

Card No. _____ Exp. Date _____

| CAT. # | PRODUCT DESCRIPTION | HOW MANY | PRICE |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | Shipping & Handling | | $1.75 |
| | Indiana residents only, add 5% sales tax | | |
| | TOTAL | | |

# PRINTER INTERFACING

**By Jack Carrel**
Staff Technical Editor

Not long after getting my computer, I would get very frustrated. It seemed that I was wasting a lot of time scrolling up and down the file to check and recheck sections of the file that were outside the 24 line window of video. I'm sure many of you have experienced this problem. Let's face it, 24 lines is not enough space to do anything significant.

There are really only two answers to this dilemma. (1) Redesign your computer and all of your software so that you can see more lines of text on your screen or (2) Get a printer. Believe me, the second choice is a lot easier on you and your bank account.

I like talking about printers, since this is one peripheral where the venerable manufacturer of our computer has allowed us an almost unlimited choice. Recently, a friend of mine asked me what would be the best choice of printer for him. Well, I learned a long time ago about giving advice on purchasing computer hardware and software. I simply do not give it. But I will gladly show someone what is available and then let them make their own decision. This usually helps a person to better live with their choice, since it really was their decision.

I decided to take my friend all over town, showing him as many different kinds and brands of printers as we could locate. But before we left, we sat down and discussed in detail what he wanted to do with his printer. Then we made a list of the things we felt he would need from a printer. By having a list of parameters with us, we would have an objective criteria whereby we could compare all of the candidates. Then we set out on our epic journey. Each time we looked at a printer we would check its specifications for those things he had decided were significant. Then we would listen to the sales person in case there was something we may have missed.

This brings us to an area of the computer industry that I feel is really lacking, i.e. salesmanship. You see, I am always seeing those wonderful advertisements on television with the cheerful, superhelpful, knowledgable salesperson efficiently demonstrating some sort of computer hardware or software. I have yet to find this paradise of computer marketing. In reality, I find stores that for the most part blend better with the image I have of the fallacious used car salesman and his lot full of questionable derelicts.

I believe the only reason that the computer industry is doing as well as it is is because consumers have such an insatiable appetite for getting involved in this field that they are willing to put up with almost anything to get what they want. This type of consumer philosophy will not last forever; as the consumer becomes more computer educated, retailers are going to have to change their attitude and policies.

During our printer trek, we encountered two major computer sales syndromes. First there was the salesman, who had convinced himself that he was literally a computer GURU. This individual would do everything within his means to communicate to us that not only was he gifted but that he was doing us a favor by even condescending to our "level". Without exception, this salestype would all but walk away when we informed him that we were going to use the printer with a TI99/4A home computer system.

The second computer sales syndrome we encountered was that of the unprepared and often tragic demonstration. Of course, I realize that any demonstration has a chance of bombing no matter what the degree of preparation is. In fact, I will admit to several such cases where I have tried to show some innocent soul a new routine, only to have my computer do something I could not even recognize. But if all I had to do was demonstrate a printer and that was my job, I would find some time to become familiar enough with it to be able to show it off.

How many of you would be willing to purchase a car from a salesman who could not get the trunk to open or figure out how to get the car into gear? I doubt if many people would purchase a car from an individual of such questionable character.

It is instances like these that make me thankful for the superior mail order services that are available to the computer consumer. In most areas of the consumer market place, the mail order business is usually considered a questionable risk. In the computer industry, not only does the mail order business usually beat the retail price, but also will deliver the merchandise before the retail store can.

I'm not saying that all retail computer stores fall victim to these sales syndromes. In fact, I am sure there are some retailers out there that have overcome these problems. It will be these businesses that will still be around when all of the computer-fed smoke has cleared and the consumers are demanding the same level of salesmanship they are demanding elsewhere. To those retailers that fall into this category, I say wait around. The rif-raf will soon fall by the wayside.

Finishing the epic tale of our recent printer conquest, my friend finally chose a printer whereupon he promptly encountered the area of computers that baffles many people — interfacing his computer to his printer.

It is not as easy as it was when you first connected your television to your computer. In fact, if you do not purchase the TI printer, the project often takes a quantum leap in apparent difficulty. I say apparent because once you have gotten a few of the facts concerning interfacing straightened out, the problems seem to fade away.

In order to best discuss printer interfacing, let's look at the basic requirements for a computer to communicate with a printer. First of all, the computer needs to send data to the printer so it can be printed. This data exchange requires that the computer and printer be connected by one or more data lines. The number of data lines depends upon the type of interface you select. Your two choices are serial and parallel.

With the parallel interface, each bit of a transmitted character has a separate line. For instance, let's say your computer transmits eight bits to a character, as the TI-99/4A does. This means that you will need eight wires between your computer and your printer. The advantage to this interface method is speed. The computer can send data to the printer at a rate of well over 10,000 characters per second. But as we will discuss later, speed of this type is usually not a significant factor for printer interfacing.

I think the parallel format offers a tremendous method of interfacing with a computer. But I feel using this type of interface with a printer really wastes its power. I prefer to use the parallel interface on my TI-99/4A with my home projects because its easier to implement and sometimes the speed is required. At a later date, I will go into more detail about parallel interfacing.

Serial interfacing sees the computer sending the data to the printer on one wire. The characters are transmitted with bits being sent one after another. So instead of the printer receiving the entire character in one unit of time, it receives each bit in a unit of time. And after a certain number of bits have been received, the printer combines these bits back into a character it can print.

There are two types of serial data transfer; synchronous and asynchronous. The difference between these two types of data transfer concerns the use of a synchronizing clock. Synchronous data transmission requires an additional clock line between the computer and the printer. This clock is used to synchronize the exchange of data between the two devices. Synchronous data transmission is generally used with very sophisticated high speed data transfer and is not available for the TI-99/4A system.

Asynchronous data transfer is accomplished without the use of a clock line. Instead, there are a set of rules that the transmitting and receiving devices must follow in order to successfully perform the data communication. These rules are necessary so the receiving equipment (in this case, a printer) will be able to distinguish one character from another.

Each character is preceeded by a start bit. This bit tells the printer that the first bit of a character is going to be on the data line during the next data unit interval. After a character has been transmitted, the next bit transmitted by the computer is called the stop bit. The purpose of this bit is twofold; to prepare the data line for the transmission of the next start bit, and to allow the receiving device time to complete the last character. The latter is a leftover of the days of the slow and cumbersome mechanical teletype machines. These machines performed all of their functions through the use of mechanical devices. They were obviously very slow.

So, the period or duration of the stop bit was used to allow the receiving device time to complete the execution of the last character and to return to a waiting state. With most printers today, the latter purpose of the stop bit is not necessary.

Now, I wish to explain a term that I have been using since its meaning is of paramount importance to serial data communication; the DATA UNIT INTERVAL. Timing is the whole key to successful serial data transfer. Since the data is transmitted one bit after another, the time in which they are received by the printer is very significant. First the start bit is transmitted. Then the first data bit is transmitted followed by the second and so on. The printer has to have some way of distinguishing between these bits. It does this by looking at the incoming signal only at certain equidistant times.

Therefore, once the start bit has been detected, the printer will know that one unit of time later the first data bit will be on the data line. And two data units after the start bit, the second data bit will be on the data line. The printer continues to sample the data line at these successive time intervals until the entire character has been received.

How does the printer know how long to wait between bits? This information is given to the printer in the form of the BAUD RATE. The baud rate specifies the number of bits per second that will be transferred between the computer and the printer during the transmission of a character. Let's say that a printer and computer are set up to transfer data at 300 baud. This means that during the transmission of a character, the bits are being transmitted from the computer and received by the printer at a rate of 300 bits per second.

More significantly, the printer will be sampling the incoming data every 3.33 milliseconds. That 3.33 milliseconds is the data unit interval. So, once the printer receives the start bit, it knows that 3.33 milliseconds later the first data bit will be on the data line, and 3.33 milliseconds after that the second data bit will be on the data line, etc.

The data unit interval is calculated by taking the reciprocal of the baud rate. So, the faster the baud rate, the shorter the data unit interval. For most microcomputers, the TI-99/4A included, the fastest baud rate is 9600. This gives a data unit interval of 104 microseconds.

The start bit and all of the data bits must be transmitted from the computer at the specified baud rate. Only the stop bit can vary from this requirement. For the stop bit, the data unit interval is the shortest duration allowed. In other words, for a 300 baud system, the stop bit interval can never be shorter than 3.33 milliseconds. But the stop bit can be, at any time, longer than the data unit interval. The purpose of this rule is simple. Suppose you wanted to type on your computer, and you wanted the characters you type to be printed on the printer. Also, let's suppose you type at about the same speed that I do, approximately 20 words per minute and we must include all errors. Otherwise, I doubt if I could even maintain that rate.

Let's say that the average word you are typing is six letters long. At 20 words per minute and six letters per word, you are typing 140 characters per minute. This includes spaces, which must also be transmitted to the printer. Now if you have your computer interfaced to your printer at 300 baud, the space between the typed characters will be approximately 460 milliseconds. This is the equivalent to the time it would take to transmit 14 characters at this baud rate.

Even at 300 baud, which is a relatively slow data transfer rate, a typist would have to type about 300 words per minute in order to send data to the printer without allowing a delay between characters. Not many of us can type at this rate. If anyone reading this can type that fast, please put down this article and drop a resume in the mail to the IUG. We could use your help immediately.

Of course, this trivial example is not the only case where the long stop bit is necessary, but let it suffice to say that there are many times when the printer needs to be prepared and waiting for the next character.

Now let's cover a subject that I feel is analogous to the myth of the 200 watt graphic equalizer for automobiles. I'm sure you have read or heard of these devices. It is my belief that you can get very superb sound reproduction and of sufficient volume without using a fraction of 200 watts.

But somewhere has evolved the philosophy that the power supplied by a sound system is directly proportional to the quality of reproduction of the sound. It is this same type of thinking that has inundated the field of computer interfacing. It seems very important to a lot of people that their printer receive data at the maximum available baud rate. As I said before, this figure stands at 9600 baud for standard serial interfaces.

This does not mean there are no cases where the maximum baud rate is necessary. But I would like to do some figuring for you.

First, let's say we have a printer that can print 90 characters per second. Given that each character consists of a start bit, eight data bits and a stop bit, this means that our printer can accept and print data from a computer at a rate of 900 baud, or bits per second. So even if we had a printer that could print at a rate of 150 characters per second, our printer could only

receive data at a rate of 1500 baud without losing or dropping characters.

Now this analysis would suggest that all you need to do to print at the faster baud rates is locate a printer that will print at the speed you will need to use the higher baud rates. Well, this is not as good an idea as it may sound. It turns out that the speed with which a printer can print characters is a very important factor in the design and technology used to build the printer. Therefore, the increase in character thoughput will give a dramatic effect upon the price of the printer. As far as I know, if you want a printer that will print characters at a rate greater than 150 cps, then you will have to spend more than $1,000 for your printer.

Don't worry, it's not as bad as it appears. You see, almost all printers come supplied with some sort of buffer. Although the size of the buffer may vary, the major purpose of the buffer does not change. A buffer in this case is a block of memory where the printer can temporarily store characters until it is ready to print them. With a buffer, the printer can receive data at a rate different from the rate at which it can print the characters. So, although the printer can only print at a rate of let's say 90 cps, it can receive data at a rate of 9600 baud.

Our problems are over, you say. We'll just make sure we get a printer that has a buffer and will accept data at a rate of 9600 baud, and there will never be a problem. Well, that's not exactly true. You see, we haven't discussed what happens when the buffer fills up. If you are only printing characters at 90 cps, then it is quite possible that you will fill the buffer very quickly.

Once the printer buffer has been filled, the printer will inform the computer by way of a control line that it can no longer accept data. In response to this information, the computer will quit sending data until it receives a signal from the printer saying it can once again accept data. Usually printers will not accept any data from the computer until the printer finishes printing the line it was on when the buffer was filled. This means that once the buffer has been filled, the computer can for the most part, only send one line at a time to the printer.

I'm sure you're beginning to see my point. After filling the buffer, the data transfer rate between the printer and the computer drops back down to the rate at which the printer can process the characters. This means that the speed advantage resulting from the use of a buffer is directly related to the size of a buffer.

Let me illustrate my point with another example. Suppose we have a printer that can print characters at a rate of 90 cps. It has a 2,000 character buffer and will accept data into the buffer at rates up to and including 9600 baud. Let's say we want to print a document with 60 spaces to a line and 5 lines to a page. This

represents a total of 3,000 characters per page. Now let's compare how long it takes to print one page at two different data transfer rates, 1200 and 9600 baud. It turns out that at 9600 baud, the printer buffer will have been filled after 2.3 seconds. At this time 2208 characters will have been transferred from the computer. From now until all of the data has been sent from the computer, the transfer rate will effectively be 90 cps or 900 baud. This data transfer will last another 8.8 seconds after the printer buffer was filled the first time. This gives us a total transfer time of 11.1 seconds.

Now let's try it at 1200 baud. At this data rate, the computer does not fill the buffer before the end of our page. Therefore, the total time to transfer the data is about 25 seconds.

Even though the higher transfer rate filled the buffer very quickly and spent about 80% of its time waiting on the printer, it still maintained a more than 2 to 1 margin over the slower baud rate.

Now let's print two pages of text and see what happens. This means we are transferring 6,000 characters to the printer. Once again, at 9600 baud, the printer buffer is filled almost immediately, in 2.3 seconds. Now the computer's effective transfer rate is lowered to that of the 90 cps of the printer. And the remainder of the transfer takes about 42 seconds. This gives us a total transfer time of 44.3 seconds.

At 1200 baud, the printer buffer once again does not get filled and the total data transfer time is about 50 seconds. As you can see, the advantage of the faster baud rate has dwindled substantially.

In fact, at three pages of text, the lower baud rate does fill the buffer and from then on the two data rates take the same amount of time.

Before this analysis, intuition would tell us that the more data you needed to print, the faster the data rate you would need. But in fact as I have shown, the data transfer rate is almost irrelevent, when you are printing large amounts of data.

Of course, there are several variables you can change that will affect the results of this experiment. If you increase the size of the buffer, the more pages of text you can print before the buffer is filled, and the more data your computer can transfer at the faster baud rate. Of course, if you increase the character thoughput of the printer, the data transfer time will be decreased for the higher transfer rate because its transfer time is dependent upon the printer speed. (80% for one page)

The reason I took the time to show you this is because I feel that you must consider both the printer's buffer size and the printer's interface data transfer rate when you are looking for a printer. Don't be tricked by media hype and high baud rates into spending more on a feature that offers no real advantage to you.

Let's get back to our discussion of the requirements of a serial asynchronous data interface system. We have already discussed the need of the data line. Also, we have been introduced to the concept of control lines. Because the printer usually cannot process data as fast as it is being received, the printer must have a line whereby it can inform the computer when it can receive data and when it cannot.

For electrical reasons, our interface also requires a ground connection between the computer and the printer. Sometimes an additional line is required by the printer in order for the computer to tell it that it is going to be sending some data.

So with these four lines our computer and printer can be fully interfaced.

This type of interfacing is a real bright spot in the field of microcomputers. As I am sure you are well aware, nothing is standard in the field of home computing. For example, a certain program will work on one computer but not on another. The same is true for disk drives and other peripherals. But in the area of serial data interfacing, there is a standard which is miraculously available on almost any home computer, the TI-99/4A included.

This interface standard is called RS232C, although usually the C is left off the name. The RS232 standard specifies both the electrical characteristics of signals and a functional description of each signal. Also, this standard describes the type of connector and pin assignments. The standard goes into much greater detail since it really covers the needs of computers to each other and to terminals. But information in this standard very conveniently covers the needs of a computer to printer serial interface.

The data from the computer is on pin 3 of the 25 pin RS232 connector. This signal is given the mnemonic of TX for transmitted data. The logic ground is on pin 7. And the signal from the printer, to signify when it can accept data, is on pin 20 of the computer connector. This signal is referred to as DTR which stands for data terminal ready.

Last of all, the signal that is sometimes required to tell the printer that the computer will be sending data, is called CTS which stands for Clear To Send. This signal is on pin 5.

I am not going to cover the pins on the printer because although they claim to follow the RS232 standard, there is often confusion on the manufacturer's part concerning what signal should go on which pin. This problem arises from the fact that several manufacturers try to match the pinouts designated for the computer while the standard does not necessarily subscribe to this philosophy.

In order to properly interface your computer and printer, you are going to have to get out the TI RS232 INTERFACE instruction manual and the interfacing instructions supplied with your printer and sit down and figure out the best configuration for your system. Often this problem is alleviated if you can locate a manufactured interface cable made specifically to work with your brand and model of printer and the TI-99/4A computer system.

I hope this explanation of the world of printer interfacing will help you not only purchase the best printer for your needs, but obtain the most from your printer.

# ASSEMBLY LINE

# GATB BREAKS THE SOUND BARRIER

**By Bill Gronos**
Senior Technical Editor
9505½ S.E. 15th #B
Midwest City, OK 73130

As the lead-in suggests, we are going to discuss the Assembly language access to the 99/4's superb sound generator. But first, how do you like $60 consoles?

I liked it so well I bought two. After all, I paid more for my Extended BASIC module! What, you may ask, am I going to do with three computers? Lots!

The most exciting idea I have is to bank two of them together via the joystick ports. Since the joystick signal levels are standard transistor to transistor logic (TTL), the coupling could be done directly and cheaply. I happen to have an old black and white TV in my garage which currently functions as a cockroach egg incubator. Imagine playing computer games with each player having his own keyboard and display screen and each player's actions are passed between the two consoles! "Global thermonuclear war" would make a good two computer game; ("Wow!, either these are the best 3-D graphics I've ever seen, or that's a real roach crawling across my screen").

If I can set the "Joy-Couple" idea to work, I'll share it with you. However, if you haven't already bought a back-up, it will probably be too late by the time you read this. Your local toy store probably sold their last one weeks ago.

Another good use for an extra computer would be for Assembly language programming support. It could be loaded with coding to do hexadecimal arithmetic, reverse assembly or many other debugging aids that now require me to interrupt what I'm currently doing in order to load the support software. The sixty bucks you spend for an extra console may well buy the best "peripheral" you can set for your 99/4.

## A MAJOR REVISION TO GATB

For any of you reading Assembly Line for the first time, GATB ™ is short for "Gronos' Assembly Translated Basic" which hopefully will aid the novice Assembly language programmer by letting him write machine language code in a Pseudo-BASIC style. Rather than redescribe the wheel in every issue of Enthusiast '99, newcomers should try to find someone with the last two issues for a further explanation,or order them through the Users-Group.

Having to continue the GATB saga from issue to issue isn't easy. Also, many of you are having problems typing in the extensive amount of coding required. I decided the best course of action would be to explain GATB in small chunks and at the end of the GATB series make entire program listings available at a nominal cost. PLEASE, DO NOT SEND EITHER MYSELF OR THE IUG ANY MONEY IN ADVANCE. I made a big mistake when I prematurely announced a booklet I was writing on Assembly language. This resulted in many readers sending in advance orders complete with checks and cash. The booklet writing came to a standstill with little hope of being finished in the near future. I will return all funds and I apologize to you all. Now that I got that off my chest, I can explain the change I am making to the GATB system.

I wasn't at all happy with the way GATB handled variables. Those of you who have been with this series from the start will remember that variables were specified as hexadecimal numbers greater than >7FFF. When a subprogram encountered a number such as >8000 (variable "0") or >8001 (variable "1") in a parameter list it would look up the variable's value in a special area of memory set aside for this purpose. This method had three drawbacks.

You couldn't give a variable a meaningful name, such as "SCORE"; negative constants were cumbersome to handle; and a special area of memory had to be pre-designated, which would tie up excessive memory unless you knew exactly how many variables you were going to need. The wasted memory was of little concern to Editor/Assembler users; they have memory coming out their ears. Minimemory users, however, scrape to save every byte they can.

Therefore, I scrapped this method and went back to the drawing board. The feature I most desired to add was to allow variable names. The conventional method would be to construct a "symbol table" such as BASIC does. The drawback to this was the waste of memory that would be caused by having to use ASCII text strings in the parameter lists. Even if we limited the variable name length to six characters, we're talking about a sizable overhead.

Then it dawned on me! Assembly language incorporates a symbol system that could be adapted to GATB with little overhead coding. I called the method "double indirect addressing". Before I attempt to explain this concept, let me review the technique of indirect addressing.

The TMS9900 microprocessor in your computer can use six addressing modes: workspace register, workspace register indirect, workspace register indirect auto-increment, symbolic memory and indexed memory. Workspace register indirect addressing is explained in section 4.1.2 in the Editor/Assembler manual. Let me clarify this for you with an illustrated example. Suppose we have a section of memory set up like this:

| ADDRESS | CONTENTS |
|---|---|
| A000 | 1234 |
| A002 | 5678 |
| A004 | ABCD |

The instructions LI 1,>A000 and MOV 1,2 would place the value >A000 into register 1 and then copy that same value into register 2. This is an example of workspace register addressing.

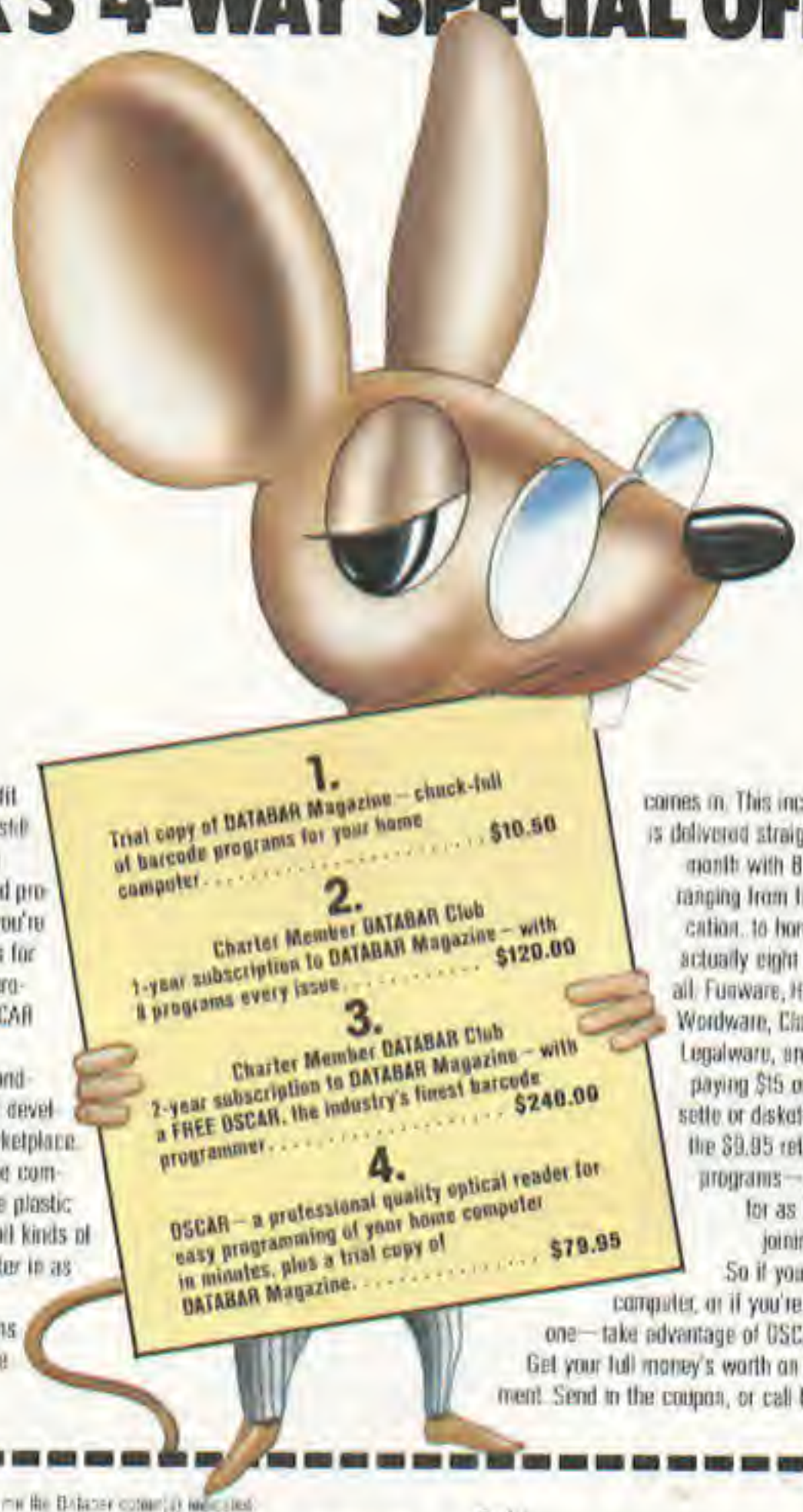
Indirect register addressing is designed by preceding the register number with a asterisk (*). The instructions LI 1,>A000 and MOV *1,2 will cause the microprocessor to use the contents of register 1 (still >A000) as an address rather than a numeric value and the value stored at memory address will be placed into register 2. So now, reg 1= >A000 and reg 2 = >1234.

If the two instructions had been LI 1,>A000 and MOV *1+,2, we would be using the workspace register indirect auto-increment form of addressing. When these two instructions are performed, reg 1 will equal >A002 (it has been auto-incremented) while reg 2 will still be given the value >1234. What would happen if a third instruction, MOV *1+,3, were executed? The result would be: reg 1= >A004, reg 2= >1234, reg 3= >5678.

What if our two instructions had been LI 1,>A000 and MOV *1,1? Register 1 would change from >A000 to >1234.

Now let me explain how double indirect addressing works in GATB. All GATB routines that duplicate BASIC subprogram calls are accessed by Branch and Load Workspace Pointer instructions. This instruction saves the address of the memory word that follows the BLWP in

# OSCAR'S 4-WAY SPECIAL OFFER!




If you want to get more benefit out of your home computer and still avoid the purchase of expensive peripherals...if complex keyboard programming leaves you cold and you're tired of paying exhorbitant prices for every new cassette or diskette program—then you're ready for OSCAR and the DATABAR Club!

OSCAR is simply the finest hand-held optical barcode reader ever developed for the home computer marketplace. It works with virtually every home computer. And together with a simple plastic template, it allows you to enter all kinds of programs into your home computer in as little as two minutes.

But hardware without programs is useless—and that's where the DATABAR Club comes in. This includes a magazine that is delivered straight to your home every month with 8 programs on subjects ranging from fun and games, to education, to home finances. There are actually eight different categories in all: Funware, Homeware, Healthware, Wordware, Classware, Scienceware, Legalware, and Genware. Instead of paying $15 or $30 or more for cassette or diskette programs, or paying the $9.95 retail price for DATABAR programs—you can receive them for as little as $1.25 each by joining the DATABAR Club.

So if you already have a home computer, or if you're thinking about buying one—take advantage of OSCAR's 4-way offer now. Get your full money's worth on your computer investment. Send in the coupon, or call for further information.

Please send me the Databar option(s) indicated.
☐ Option 1 ($10.50) ☐ Option 3 ($240.00)
☐ Option 2 ($120.00) ☐ Option 4 ($79.95)*
*30 day money back guarantee

Enclosed Check # _____

Money Order # _____
Payable to Databar.

Or charge my Am Ex ☐ Visa ☐
MasterCard ☐ Diners Club ☐
Carte Blanche ☐

Card # _____

Exp. Date _____

Name _____

Address _____

City _____ State _____ Zip _____

My Computer Model is _____

**DATABAR**

Databar Corporation, 10202 Crosstown Circle • Eden Prairie, MN 55344 • Phone Orders: MN (612) 944-5700
Out-of-State: 1 (800) 572-2776

register 14 of the subroutine workspace. The GATB routines know where to set the parameter list from because the address of the first parameter is saved in reg 14. We'll analyze the double indirect addressing method by seeing how the GATB SOUND routine fetches the value of the sound duration variable named "TIME". The coding might look like this:

```
                        DEF RUN
A000 03E8  TIME         DATA 1000
A004 006E  FREQ         DATA 110
A006 0002  VOLUME       DATA 2
A008 0000  ZERO         DATA 0
A00A 0420  RUN          BLWP@SOUND
A00C A100
A00E A000               DATA TIME,FREQ
A010 A004
A012 A006               DATA VOLUME,ZERO
A014 A008
A016 10FF               JMP $
```

Time for a pop quiz: what value will register 14 of the SOUND subroutine have when the BLWP @SOUND is executed? Reg 14 will equal >A00E because the BLWP @SOUND requires two words of memory and >A00E is the first word of memory following the instruction. Did you say ">A00C"? If so, "GOT YA"!

Now let's take a look at the SOUND routine instructions that fetch the first parameter in the data list, which happends to be the duration value. The parameter list *does not* contain the actual duration value; it has the address of the memory word that stores that value. The first part of the fetch is MOV *14,4. This loads reg 4 with address >A0000 and increments reg 14 to >A010. The second part of the fetch is the instruction MOV *4,4. Finally, reg four contains >03E8. It took two indirect instructions to do this, hence the name "double indirect addressing".

Thus, this method lets us use variable names that can have a meaning to us, but it does have a drawback: we can't use constants in the parameter lists. That was why symbol ZERO had to be specified. If instead of the label ZERO we had put numeral 0 for the last value of the parameter list, the value stored at address >0000. which sould be >83E0 no matter what version of 99/4 or 99/4A you have because >0000 and >0002 are the context vectors for the reset interrupt, would have been used by the SOUND routine. That would have caused some pretty weird sounds and would probably "lock up" your console.

## THE GATB SOUND ROUTINE

I still get many letters telling me I'm writing above the experience level of some readers. As I've said before, my audience is too diverse to permit a median level of experience to be assumed. This month's ASSEMBLY LINE is going to bore all you more experienced Assembly language programmers because I'm going to cover a lot of elementary points such as hexadecimal conversion and "twos complement" arithmetic.

I have written extensive notes within the coding for the GATB SOUND routine. Because of this, the program list is printed at the end of the article without the notes so that it can be entered more easily.

**Minimemory owners will need to limit the length of the labels to two characters.**

This is a "stand alone" routine and does not require any of the previous GATB coding given in the past issues of ENTHUSIAST '99. The parameter list that **must follow the BLWP@SOUND instruc**tion has the same order as the Basic CALL SOUND subprogram: duration, freq 1, volume 1, freq 2, volume 2, freq 3, volume 3, noise type, noise volume. You can specify all three tones and one noise at the same time or you may omit any of them. However, you must put a variable name that has a value of 0 at the end of the parameter list so the sound routine will know when to quit.

The following sound calls are all valid:

```
BLWP @SOUND
DATA TIME,FREQ,LEVEL,ZERO
BLWP @SOUND
DATA X,Y,Z,ZERO
BLWP @SOUND
DATA D,F1,V1,F2,V2,ZERO
LI 1,1000
MOV 1,@TIM
LI 2,3000
MOV 2,@FREAK
LI 3,4
MOV 3,@VOLUME
CLR @END
BLWP @SOUND
DATA TIM,FREAK,VOLUME,END
```

Not legal is:

```
BLWP @SOUND
DATA 1000,110,2,0
```

Of course, no error statement will be issued, but you're going to hear some mighty strange sounds!

## GATB CODING COMPLETE WITH THOROUGH DOCUMENTATION GATB EQUIVALENT OF BASIC CALL SOUND

```
SUBWS1 BSS 32
```

A register area is allocated for use by the SOUND subroutine.

```
SNTBPT EQU >83CC VDP
   TBL LOC
SNSTRT EQU >83CE LOAD
   Ø1 TO START
SNTBLC EQU >83FD SET
   LS BIT
```

We EQUate symbolic names for three "scratch pad RAM" addresses that are used by the automatic sound routine. The symbols are abbreviations for SOUND TABLE POINTER, SOUND START and SOUND TABLE LOCATION.

```
SNDOFF DATA >Ø49F,>BFDF,
   >FFØØ
```

This data string contains four sound operations that specify maximum attentuation values for the tone generators and the noise generator. These operations will be appended to the end of every sound list so that the sound will stop.

```
SOUND DATA SUBWS1
     DATA $+2
```

This is the standard heading for a subroutine that is to be accessed by a Branch and Load Workspace Pointer (BLWP) instruction. SOUND is the name of the routine, SUBWS1 is the label that marks the beginning of the workspace area (which is defined at the end of the coding). $+2 is the location of the first instruction. "$" is the shorthand symbol for the current memory location, so +2 will be the next available word of memory. We could have used DATA SUBWS+4, which indicates the same memory address, but requires more typing. These two lines could be merged into: SOUND DATA SUBWS1,$+2 which requires even fewer keystrokes and a few less bytes of space when the coding is saved to disk.

In the Editor/Assembler manual, a heading such as this is called a "context vector".

```
LI 6,>3ØØ1
LI 5,>8ØØØ
```

All the instruction bytes that are fed into the sound generator require that the

most significant bit be set to one. Hex >8000 only has this bit set and will be used in formatting the sound instruction bytes.

MOV *14+,4

We fetch the first variable address from the string of names following the BLWP @SOUND instruction. The BLWP instruction automatically saves the address of the next program step to be executed when the subroutine call is completed and return to the main program is executed with a ReTurn with Workspace Pointer (RTWP) instruction. In this use, however, the BLWP is followed by a data string that lists the parameters of the sound list.

Since GATB attempts to emulate Basic functions as nearly as possible, the sound list follows the same format that would follow a CALL SOUND instruction: duration,tone no. 1 frequency, tone no. 1 volume, tone no. 2 frequency, tone no. 2 volume, tone no. 3 frequency, tone no. 3 volume, type of noise, volume of noise. The first parameter is the duration we want the sound to be heard specified in milliseconds. Actually the sound routine doesn't come anywhere near this precision.

A millisecond is 1/1000 (.001) part of a second. Did you ever notice that if you specify either 1 or 10 milliseconds as the duration in a BASIC CALL SOUND, both values produce tones of the same length? This is not because the sound processing IC chip is inprecise, it's because Basic is using the built in interrupt routine to control the sound time duration. The interrupt concept was explained in detail in the November Enthusiast '99, so I'll only briefly recap the highlights.

The standard system interrupt in the 99/4 is triggered by the video synchronization signal that is fed to your monitor to stabilize your screen. It occurs 60 times per second. This 1/60 second spacing is critical, so the Video Display Processor IC chip must have a separate time base circuit. All computer timing circuits are based upon a "crystal" which oscillates at a very high frequency in order to achieve a very high precision.

Remember a number of years ago, long before digital electronics exploded into home use, when the Accuratron watch was first introduced? We were bombarded by commercials touting the extreme accuracy of this watch which used a tiny tuning fork to break a second into 120 parts. Ordinary wristwatches used a ratchet driven escape movement to separate a second into five pieces. Thousand dollar Swiss watches had very precise movements that could cycle in 1/20 second.

Well, along came the digital revolution and Swiss watchmakers looked on in horror as ugly, impersonal factories spewed out watches by the thousands which were so accurate the Swiss chronographs looked like hour glasses in comparison. Soon, everyone and their dog had watches that were accurate within minutes per year.

But using a 120 vps tuning fork to time a high speed digital computer would be like using fireplace tongs to splice genes. Thus, it is a tiny sliver of quartz crystal that vibrates 48,000,000 times per second which provides the critical timing signals used by your computer's logic circuits. This clock speed far exceeds the maximum 3.33 megahertz clock rate that the TMS 9900 microprocessor can handle, so it is "downlocked" to a moderate 3 mhz.

This highly accurate timepiece described above is the master clock for your system. It is this clock speed that determines how long a given machine language instruction will take to execute. However, this clock speed is useless to your video display processor chip circuits.

Due to the frequency requirements of your monitor, the video processor has to have its own peculiar clock signal which is generated by a separate timing network. Again, a slice of quartz initiates the clock cycle, but this one has the seemingly odd frequency of 3.579545 megahertz. This is the clock that controls the color level of your monitor, the individual frequencies that can be produced by your sound generator and, last but not least, that very important 1/60 second interrupt signal.

I hope those of you who write me asking that I put more explanatory notes with the program coding are happy; the notes are longer than the instruction list. Now I can get back to telling you why a 1 millisecond duration sound is the same as a 10 millisecond note.

Since the interrupt routine occurs every 1/60 of a second, this determines the "resolving power" of all the functions that are interrupt driven: Sprites have speeds that are multiples of the fundamental 60 hz cycle, the "screen time out counter", which protects your monitor picture tube phosphers, blanks the screen after 32,768 sixtieths of a second have passed without a key being pressed and, as you may have guessed by now, the shortest period of time a tone can be burned on is 1/60 second.

One sixtieth of a second= 16.667 milliseconds, ergo a 1 millisecond tone will last just as long as a 10 millisecond tone or a 15 msec one. The tone must be turned **on during an interrupt cycle and the quickest it can be turned off is during the next interrupt.**

While this disparity seems quite large for a very short tone, it is negligible for tones of 1/4 second or longer, with a maximum error of 6.7%.

Since this is a tolerable margin of error, there's no reason why we should trouble ourselves with writing a sound routine that is more accurate than that which is provided in BASIC. If we did, program instructions translated from Basic into GATB would sound different when the specified duration is very short. Let's continue on to the next instruction.

MOV *4,4

Now we have the actual sound duration stored in register 4.

JGS SOUND3

Here we test to see if the duration is negative. If so, we will halt any sound that is already in progress and begin the new sound.

"JGT" is the Jump if Greater Than instruction. There are a dozen different "Jump" instructions. These are used as decision making points where alternate paths are chosen depending on conditions existing when the conditional jump instruction is executed.

You can pair off the jump instructions with the compare instructions (Compare Immediate,CI; Compare words,C; Compare Bytes,CB) to duplicate the BASIC If . . . Then instruction. In this case it isn't necessary to do a comparison because the Move instruction (MOV) has already done this.

Like many other Assembly language instructions, when a Move instruction is completed the microprocessor will perform an automatic comparison between the value being moved and zero. This is just what we wanted, because sound durations of less than zero require immediate execution and those greater than zero will "mark time" until the previous sound is completed.

Now we come to a point that many people have trouble understanding: how negative values are handled in binary arithmetic.

Negative binary values are expressed as "twos complements" of the corresponding positive numbers. This is easier to understand if you think of it as "odometer" arithmetic. If a six digit odometer was set to zero and then turned backwards one division, it would read "999999". This would be the "tens complement" of the number 1. In effect, this is how the odometer would register negative values because now it will again register zero when a positive value of one is added to it by turning it forward one division(—1+1=0).

While "tens complement" does in effect allow us to have negative values without having to make room for a minus (—) sign, it can cause confusion: the odometer will register "999999" when it is turned forward that number of miles or if it is turned backwards one mile. This is an important difference, because we all would prefer to have a car that was rolled backward one mile rather than one which has been driven nearly four times around the world.

To fit this odometer example to computer math, we will have to imagine a four digit odometer with wheels that are marked 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F. These are the sixteen single digit hexadecimal numbers. Each digit requires four binary bits, so the largest value that can be represented by a word length of sixteen bits is >FFFF (the " "symbol means we are indicating a hexadecimal value), which equals decimal 65535. All digital computers use binary number representation, which is also called "base two" since all numbers must be expressed with only two symbols: 0 and 1. Hexadecimal numbers are also called base sixteen because that is how many symbols can be used for a single digit. Hexadecimal notation is used because it only requires four digits to represent a word value rather than a binary string that requires sixteen "ones and zeros".

Just like the automobile odometer, there is no direct way to express negative numbers. The same "roll backwards" method can be used to change positive numbers to their negative values and this is called the "twos complement": -1 =>FFFF -2=>FFFE, -3=>FFFD, etc. This rolling backwards of a symbolic odometer may be alright for finding the negative equivalents for small numbers, but is there an easier way for larger numbers? What is the twos complement of —1000?

First we have to convert +1000 to its hex equivalent. Special calculators are available that will aid you in this task (one such model is called the "TI PROGRAMMER"), but an ordinary $5 calculator will do the same trick if you don't mind punching in a bunch of extra keystrokes. Here is a step by step procedure:

Step #1: become familiar with the "place values" for each of the four digits in a hexidecimal number; 4096, 256,16,1.

Step #2: write down a zero for each of the place values that is bigger than the decimal number you wish to convert. Using 1000 as an example, 4096 is the only value that is larger, so we mark down one zero.

Step #3: divide the decimal number by the first place value that is smaller: 1000/256= 3.90625. Write down the whole number portion of the answer after the zeros you obtained from step #2. Still using 1000 as an example: 0,3.

Step #4: subtract the whole number portion from the division result (3.90625—3= .90625). Multiply the remaining decimal portion by 16(.90625 = 14.5) and again write down the whole number portion after the other values you obtained (0,3,14).

Step #5: repeat step #4 until you get four values written down (14.5—14= .5;

Step #6: substitute the hexadecimal value for each of the four values (0=0; 1 =1; 2=2...9=9; 10=A; 11=B; 12=C; 13=D; 14 =E; 15 =F, >OE38. The conversion is complete: 1000= >OE38.

Now you can see why an extra 99/4 console comes in handy for program support. You can be using one console to write the program using the Editor/Assembler or Minimemory and you can have the other console programmed with support software such as a Basic decimal to hexadecimal converter and vice versa. Let's do another conversion step by step to make sure everyone understands it. We'll convert 99:

Step #2 result: 0,0.

Step #3: 99/16=6.1875, 0,0,6

Step #4: 6.1875—6=.1875, .1875=3, 0,0,6,3.

Step #5: not required since we already have four values.

Step #6: >0063. Voila!, we have the answer. Gee, this is kind of fun; let's do one more: 25000.

Step #2: no zeros.

Step #3: 25000/4096=6.1035156. 6.

Step #4: 6.1035156—6=.1035156,

Step #5: 1.6562496—1=.6562496,

Step #5 (repeat): 10,499993—10 =.499993, 8), 6,1,10,8.

Step #6: >61A8. 25000= 61A8.

Now that you can convert positive numbers into their hexadecimal equivalents, let's learn to convert negative numbers. What is 25000 in hex?

Take the four values you ended up with from step #5 of the above conversion (6,1,10,8) and subtract each of them from 15: 9,14,5,7. Perform step #6 for these values: >9D57. Add 1 to this result: >9D58 =—25000.

For those of you who are skeptical of this method, test it out for yourself. Any negative number when added to its absolute value will equal zero: —25000+25000 =0. Let's add the hex equivalents of these two numbers and see if this rule of additive inverses" holds true: >9D58+>61A8 =>10000. What happened? >10000 doesn't equal zero, it equals 65536!

Actually, in 16 bit binary arithmetic >10000 does equal zero because this number requires 17 bits and 17 bits just isn't going to fit in one word of memory. And no, that 17th bit isn't going to "carry over" to the adjoining word of memory, it is ignored. This is just when your car odometer hits 100,000 miles; it only shows the five zeroes.

This method for handling negative numbers in binary works well, but it does pose one ambiguity: many binary numbers are going to have two possible values. If we only consider the values as positive numbers, sixteen binary digits will allow us a range of numbers from 0 to 65535. However, if we consider all that have the most significant bit (the leftmost bit) to be negative numbers, the range of values becomes —32768 to +32767. Therefore, >FFFF can be either a whopping 65535 or a miniscule —1. How does the computer know which value we wish to use? — we tell it.

And how do we tell it? By choosing the appropriate conditional jump instruction depending on whether we want a logical or an arithmetic comparison. Arithmetic comparisons take the sign of a number into consideration but logical comparisons do not. So when we say

arithmetic, we are referring to a range of numbers between —32768 to +32767. Logical, conversely, treats all numbers as members of the set from 0 to 65535.

There are four logical conditional jumps: Jump if High (JH); Jump if High or Equal (JHE); Jump if Low (JL); Jump if Low or Equal (JLE), but there are only two arithmetical jumps: Jump if Greater Than (JGT); Jump if Less Than (JLT).

Here are several comparisons to illustrate the difference between logical and arithmetic comparisons. "Logically greater than" is abbreviated lgt and "arithmetically greater than" is agt:

1 is lgt and agt 0.
25000(>61A8) is lgt and agt 0.
40280(>9D58) is lgt 25000.
25000 is agt 40280(hex 40280 = —25000).
32768(>8000) is lgt 32767(>7FFF).
32767 is agt 32768.
0 is agt 32768.

So with arithmetic comparisons, >0— > 7FFF are considered to be larger than > 8000— FFFF.

Getting back to our sound duration parameter, you should now see how we determine if the specified interval is negative or positive. The MOV *4,4 automatically causes the new register 4 value to be compared with zero and negative numbers are less than zero. The JGT SOUND instruction will transfer program to the first instruction following the label SOUND. Since the largest duration value that is allowed in the BASIC CALL SOUND instruction is 4250, which is never treated as a negative number in hex, we don't have to worry about a long duration causing an incorrect jump. Next instruction:

```
ABS 4
```

Now that the negative/positive test has been made, we use the absolute value instruction to change the negative value to positive. Since register 4 will never contain a positive value at this point, we could also have used NEG 4, which would have nested the value. Remember, a negative of a negative number is a positive number.

```
SOUND2 MOV *14+,Ø
```

This is the point in the program road where the negative and positive travelers meet. The negative duration sets here directly, but the positive value reaches here after a check was made to see if another sound instruction was still in progress.

This instruction fetches the next parameter value from the date string following the BLWP @SOUND instruction. This parameter will be a frequency specification variable address.

```
MOV *Ø,Ø
```

We dump the variable address out of register 0 and replace it with the actual frequency value.

```
JEQ SOUND1
```

Another automatic comparison with zero was made in the previous instruction. This was to test to see if we had come to the end of the sound data string. If the frequency equals zero, program control will transfer to label SOUND 1.

```
CI Ø,>FFF7
```

How well did you learn the two's complement lesson? Here's your chance to get a little practice. What negative decimal number does this hex value equal? We find the value that is needed to make each digit add up to 15, which is 0008, and then we add 1 and to get the answer: —9. Can any of you guess what the purpose of this instruction is? We are testing the frequency to see if it's a tone or a noise. Noises are specified by a negative numbers from —1 to —8.

```
JH NOISE 1 #
```

Jump High (JH) is a logical test, not arithmetic. Therefore, the only values that will cause the jump to be made are the eight noise values —1 to —8. All other values will be construed to be tones and program control will continue on to the following instruction.

```
LI 1,1
LI 2,>B4F5
```

These two instructions are loading the frequency constant >1B4F5, decimal

value 111861, into two consecutive registers. One register is one word of memory, so the largest value a register can hold is 65535. Where did I ever get that number from? Editor/Assembler section 20.1.2-frequency specification.

```
DIV Ø,1
```

We divide the tone value into the frequency constant to derive the value that must be loaded into the tone generator to produce the specified tone. Lucky for us, the Divide (DIV) instruction can handle a two word dividend (no, not a stock dividend; the number we are dividing into). A frequent mistake made by beginners is to forget to clear the unused register when the dividend is only one word. The Divide instruction treats the "designation operand" (in this case register 1) as a two word value. This particular usage will divide the two-word value contained in registers 1 and 2 by the value of register 0. The result is placed into registers 1 and 2 with the whole number going in register 1 and the remainder going in register 2.

What are the lowest and highest frequencies the tone generator can produce? The Editor/Assembler manual says 110 and 55,938 hertz (one hertz= one cycle per second). Note that this is a greater range than with BASIC, which allows values between 110 and 44,733 hertz. I pondered this difference for fifteen minutes and couldn't think of a reason why BASIC doesn't support the higher freqs; Another 99/4 enigma.

```
MOV 1,2
```

Since we don't care about the remainder, we can use register 2 to keep a copy of register 1.

```
ANDI 1,>F
```

We only want the four least significant bits, so this And Immediate instruction will mask out the other twelve. A tone frequency is fed into the sound generator as two bytes. It is the first byte that carries the four least significant bits.

```
SWPB 1
AB 5,1
```

The SWaP Bytes (SWPB) instruction exchanges the positions of the least and most significant bytes of register 1. Add Bytes (AB) merges the four frequency bits in positions 4-7 (0 being the most significant bit) with the four bits in positions 0-3 that specify the operation. Register 1 now contains the binary string 1000XXXX0000 0000000000(bits represented by X's will vary with the specified frequency).

```
AI 5,>1ØØ2
```

This instruction increments the most significant byte of register 5, which effectively makes it the code for the next sound operation. This allows the building of the sound list to be condensed into an efficient loop. We are going to make register 5 pull double duty; the right-hand byte will serve as a counter for the length, in bytes, of the sound list. The length will vary with the number of tone generators that will be activated by a given instruction.

```
MOV 6,Ø
```

At the start of this sound routine we initialized register 6 with the value >3001. This is the location in VDP RAM where the first sound instruction will be placed. There is nothing special about this location and any other unused area of VDP RAM would work equally well.

We transfer the desired VDP address into register 0 in preparation for the Video Single Byte Write (VSBW) subroutine call.

```
AI 6,3
```

This updates register 6 with the next available address in the VDP RAM sound list. It is incremented by three because it takes two bytes to specify a frequency and one byte to specify the atenuation.

```
BLWP @@VSBW
```

We shoot the byte into VDP RAM.

```
MOV 2,1
```

Register 2 is where we placed a copy of the frequency code. Now we copy it back into register one so we can build the second part of the frequency instruction.

```
SRA 1,4
```

We already took care of the four least significant freq code bits, so we shift them off into the twilight zone via a Shift Right Arithmetic (SRA).

```
SWPB 1
ANDI 1,>3FØØ
```

We build part two of the frequency specification instruction.

```
INC Ø
```

We advance the VDP RAM address one byte.

```
BLWP @@VSBW
```

Off it goes into the VDP memory.

```
MOV *14+,1
```

Another variable address is plucked from the parameter list.

```
MOV *1,1
```

Again, the variable address is replaced with the variable value.

```
SWPB 1
AB 5,1
```

The operation code is added on.

```
AI 5,>1ØØ1
```

Increment both the OP code and the sound list length.

```
INC Ø
```

Next VDP address.

```
BLWP @@VSBW
```

There she goes.

```
JMP SOUND2
```

We return to the start of the tone instruction building loop and do it again.

```
SOUND1 MOV 4,1
```

Do you remember what we stored in register 4 way back when? Right, the sound duration. We move it into register 1 so it can be sent to VDP memory.

```
SRL 1,4
```

Here I cheat a little. The duration is given in milliseconds, but the automatic sound processing routine works in 1/60th second intervals, not 1/1000ths. To convert the mills into sixtieths requires we divide the mills by 16.667. Since we can't work with decimal points, the exact way to handle this would be to multiply the duration by 1000 and then divide it by 16667.

Multiply and divide instructions take lot'sa time. Instead, we sacrifice a little accuracy and divide by an even 16. "Wait a minute", you say, "that's not a divide instruction?" Oh yes it is! We can multiply and divide by powers of two by shifting a register left or right respectively. The number of bits shifted is the power of two we are multiplying or dividing by. In this case, 2 to the 4th power equals 16, so register 1 does get divided as required. The amount of error works out to about 4%. The sound duration will be increased by that amount. Can your ears tell the difference between 60 milliseconds and 62 milliseconds? I rather doubt it.

```
SWPB 1
MOV 6,Ø
BLWP @@VSBW
```

The duration is tacked on to the end of the sound list.

```
INC Ø
LI 1,SNDOFF
LI 2,6
```

```
BLWP @@VMBW
```

Last of all, we add the sound instructions needed to turn off the tone and noise generators. If these were not added, the automatic sound routine would quit, but the last sound list processed would remain on until another BLWP @SOUND was executed.

```
LI Ø,>3ØØØ
SWPB 5
MOV 5,1
BLWP @@VSBW
```

We place the length of the sound list in the first byte of the VDP buffer.

```
MOV Ø,@@SNTBPT
```

This tells the autosound routine where our sound list is.

```
LI 1,>1ØØ
SOCB 1,@@SNTBLC
MOVB 1,@@SNSTRT
```

We set a bit that tells the autosound routine our sound list is in VDP RAM. I guess this means we could put the sound list into CPU RAM by setting a different bit or resetting this one. I dunno, I never tried it. It would waste a little CPU RAM (and remember, when your making modules you have precious little of that) but the sound routine would execute faster. Since we're limited to a resolution of 1/60 of a second, it probably wouldn't make any difference. After all, a sixtieth of a second in Assembly language is like a geologic era to mankind (that's peoplekind for all you libbers).

```
RTWP
```

Back we go to the calling program. Did I forget to turn interrupts back on with a LIMI 2 instruction? Nope. The RTWP will return the status register to the state it was in prior to the BLWP. The status register contains the interrupt mask, so whatever level of interrupts was allowed prior to the BLWP will be in force after the return.

```
SOUND3 MOVB @@>83CE,1
```

This is where the program goes if a positive duration was specified. It checks to see if a previous sound list is still being executed. If Address >83CE equals zero, no sound is being executed.

```
JNE SOUND3
```

If there is a sound in progress, we will loop until it finishes.

```
LIMI Ø
JMP SOUND2
```

The previous sound has ended. Turn off the interrupts (because we will be accessing VDP memory) and begin processing of the sound parameter list.

```
NOISE 1 ABS Ø
 DEC Ø
 ANDI Ø,7
 AI Ø,>EØ
 SWPB Ø
 MOVB Ø,1
 MOV 6,Ø
 AI 6,2
 INCT 5
 BLWP @@VSBW
 MOV *14+,1
 MOV *1,1
 SWPB 1
 AI 1,>FØØØ
 INC Ø
 BLWP @@VSBW
 JMP SOUND 2
```

This sequence creates the sound list instruction needed to process a noise in a similar manner by which the tone instructions were created.

```
END
```

Don't forget that if you want this sound routine to work you must enable interrupts with LIMI 2.

**P.S.**

I will no longer be writing articles on the CC-40 Compact Computer. The CC-40 I had was donated to the User's Group by TI.

*(program listing on next page)*

---

*MEMBER SPOTLIGHT CONTINUED . . .* from page 10)

"99 LINES". Our group now has about 200 members and is growing rapidly.

As I reflect back on several years of TI-99/4 and 99/4A use, I have many fond memories of the hours spent entering and running programs. It is amazing how much you can learn about a computer system just using it, even without taking any formal classes. We have added the Expansion System to replace our original "train" to the right of the TI-99/4A, and an Okidata 92 printer zipping along at 160 CPS has replaced our old clunker which barely typed 10 WPM when it was working properly. The TI-Writer word processing program has made typing a dream for my wife and me. I don't know how we got along without it. My son is using Logo II and other educational programs as much as other games he plays. We inventoried recently and found that we have downloaded over 80 programs from the TexNet-Source files donated by the IUG and other individuals. Winston-Salem has three bulletin board systems with no connect charge. Someday we would like to have one on a TI-99/4A too.

Recently I have interfaced my TI-99/4A to my Kenwood ham rig and can now send or receive Morse code, radio teletype, or ASCII and see all signals on the screen or output them to my printer. Look for me on the 40 meter ham band. My call is N4QG.

Athene has been teaching part-time at the Winston-Salem TI Learning Center as well as teaching full time at a local community college. At present she is learning Multiplan so that we can project our family tax bill more accurately.

Our whole family is excited by the recent purchase of a Texas Instruments Professional Computer which is the latest addition to our computer room. This makes us a member of the growing number of two computer families. Now we don't have to store our work when the neighborhood kids come over to try out the latest games.

I am a little envious of all the new TI-99/4A owners who paid less than $100 for their computer. Still for those who ask me if it was worth the high price to be an earlybird home computer owner I can truthfully respond "You bet your push buttons!"

```
0001  *SOUND IMITATES BASIC SUBPROG
0002  *PUT A ZERO AFTER LAST ATTEN
0003  *USES VDP MEM AT >3000
0004
0005   REF VSWB,VMBW
0006
0007  SUBWS1 BSS 32
0008  SNTBPT EQU >83CC VDP TBL LOC
0009  SNSTRT EQU >83CE LOAD 01 TO START
0010  SNTBLC EQU >83FD SET LS BIT
0011  SOUNDK DATA >B4F5
0012  SNDOFF DATA >049F,>BFDF,>FF00
0013
0014  SOUND DATA SUBWS1,$+2
0015   LI 6,>3001
0016   LI 5,>8000
0017   MOV *14+,4 DURATION
0018   MOV *4,4
0019   JGS SOUND3 IF POS,WAIT TIL LAST SND END
0020   ABS 4
0021  SOUND2 MOV *14+,0 FREQ
0022   MOV *0,0
0023   JEQ SOUND1
0024   CI 0,>FFF7
0025   JH NOISE 1 #
0026   LI 1,1
0027   LI 2,>B4F5
0028   DIV 0,1
0029   MOV 1,2
0030   ANDI 1,>F
0031   SWPB 1
0032   AB 5,1
0033   AI 5,>1002
0034   MOV 6,0
0035   AI 6,3
0036   BLWP @@VSBW
0037   MOV 2,1
0038   SRA 1,4
0039   SWPB 1
0040   ANDI 1,>3F00
0041   INC 0
0042   BLWP @@VSBW
0043
0044   MOV *14+,1 ATTEN
0045   MOV *1,1
0046   SWPB 1
0047   AB 5,1
0048   AI 5,>1001
0049   INC 0
0050   BLWP @@VSBW
0051   JMP SOUND2
0052  SOUND1
0053   MOV 4,1
0054   SRL 1,4
0055   SWPB 1
0056   MOV 6,0
0057   BLWP @@VSBW
0058   INC 0
0059   LI 1,SNDOFF
0060   LI 2,6
0061   BLWP @@VMBW
0062
0063   LI 0,>3000
0064   SWPB 5
0065   MOV 5,1
0066   BLWP @@VSBW
0067
0068   MOV 0,@@SNTBPT
0069   LI 1,>100
0070   SOCB 1,@@SNTBLC
0071   MOVB 1,@@SNSTRT
0072   RTWP
0073  SOUND3 MOVB @@>83CE,1
0074   JNE SOUND 3
0075   LIMI 0
0076   JMP SOUND2
0077  NOISE 1 ABS 0
0078   DEC 0
0079   ANDI 0,7
0080   AI 0,>E0
0081   SWPB 0
0082   MOVB 0,1
0083   MOV 6,0
0084   AI 6,2
0085   INCT 5
0086   BLWP @@VSBW
0087   MOV *14+,1
0088   MOV *1,1
0089   SWPB 1
0090   AI 1,>F000
0091   INC 0
0092   BLWP @@VSBW
0093   JMP SOUND 2
0094   END
0095
```

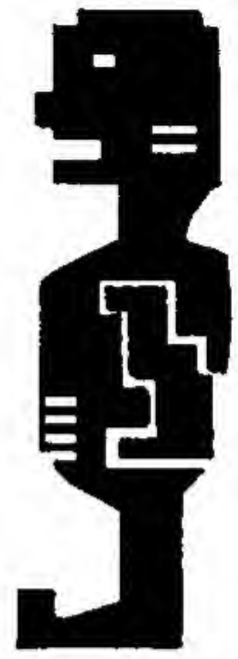
```
10 REM   CREATED    BY                  SAM MOORE JR.
20 REM    SHERMAN, TEXAS                    75090

30 REM   ENTHUSIAST '99                  VERSION

40 CALL CLEAR
50 FOR CH=100 TO 130
60 READ A$
70 CALL CHAR(CH,A$)
80 NEXT CH
90 CALL COLOR(9,5,1)
100 CALL COLOR(10,5,1)
110 OPEN #1:"SPEECH",OUTPUT
120 PRINT #1:"//37"
130 R=1
140 C=2
150 A$="HI. I AM ROCKY ROBOT."
160 B$="THIS PROGRAM IS ABOUT..."
170 C$=""
180 D$=""
190 E$="     LEARNING DIVISION"
200 F$=""
210 PRINT :::
220 GOSUB 660
230 CALL CLEAR
240 AW1=INT(RND*99)+1
250 AW2=INT(RND*99)+1
260 ANS=AW1/AW2
270 Z=ANS-INT(ANS)
280 IF Z<>0 THEN 240
290 GOSUB 980
300 PRINT "WHAT IS";AW1;"DIVIDED BY";AW2;"...":::
310 GOSUB 850
320 GOSUB 950
330 GOSUB 1170
340 GOSUB 950
350 GOSUB 1100
360 GOSUB 950
370 A=0
380 CALL CHAR(101,"000F")
390 CALL KEY(0,K,S)
400 IF S=0 THEN 390
410 CALL SOUND(-100,880,29)
420 PRINT #1:CHR$(K)
430 IF K>47 THEN 460
440 CALL SOUND(200,110,0)
450 GOTO 390
460 IF K>57 THEN 440
470 A=A+1
480 IF A<>1 THEN 510
490 A$=CHR$(K)
500 IF ANS<=9 THEN 570
510 IF A<2 THEN 390
520 IF A=3 THEN 560
530 A$=A$&CHR$(K)
540 IF ANS<=99 THEN 570
550 IF A<3 THEN 390
560 A$=A$&CHR$(K)
570 IF ANS<>VAL(A$)THEN 610
580 PRINT #1:"THAT IS CORRECT"
590 GOSUB 1430
600 GOTO 230
610 PRINT #1:"THAT IS INCORRECT"
620 GOSUB 1530
630 GOTO 230
640 GOSUB 660
650 END
660 REM
670 CALL CHAR(101,"000F")
680 PRINT :::::A$:B$:C$:::D$::E$:F$::::
690 GOSUB 850
700 GOSUB 950
710 PRINT #1:"HI. I AM ROCKY RO-BOT"
720 GOSUB 950
730 PRINT #1:"THIS PRO-GRAM IS ABOUT"
740 GOSUB 950
750 PRINT #1:C$
760 GOSUB 950
770 PRINT #1:D$
780 GOSUB 950
790 PRINT #1:"LEARNING DIVISION"
```

# LEARNING DIVISION

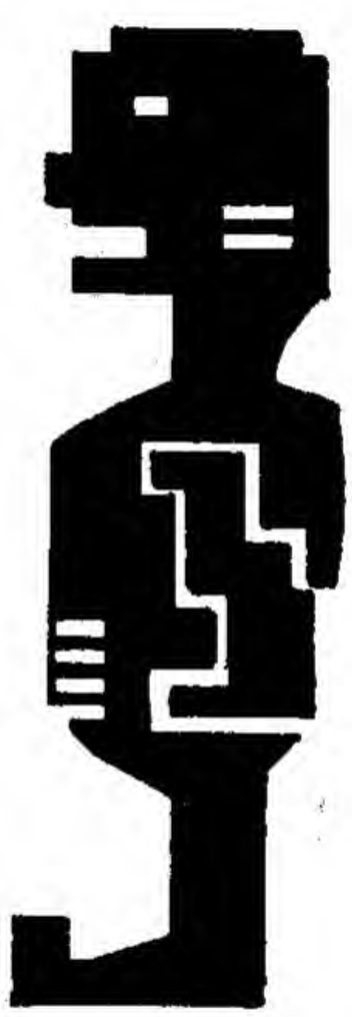
To those of you who have been members of the IUG for a year or more, the name Sam Moore, and his famous character, Rocky Robot, are quite familiar. Rocky tells jokes and stories, helps children learn mathematics and fundamental reading skills, and has become an integral part of the IUG's Software Exchange Library.

The program listing in this issue of Enthusiast '99 is Learning Division. Once typed in, it can be saved to either disk or cassette and requires the use of PHM 3035, Terminal Emulator II, and Speech Synthesizer. Learning Division was developed for students seeking to enhance their skills using a drill and practice session with the 99/4A Home Computer. Once mastered, the program can be easily changed to accomodate more advanced math levels.

```
800 GOSUB 950
810 PRINT #1:F$
820 CALL CHAR(101,"000F")
830 GOSUB 1050
840 RETURN
850 REM
860 FOR RR=18 TO 23
870 FOR CC=29 TO 30
880 PP=RR
890 IF CC<>30 THEN 910
900 PP=RR+6
910 CALL HCHAR(RR,CC+1,82+PP)
920 NEXT CC
930 NEXT RR
940 RETURN
950 CALL CHAR(101,"000F")
960 CALL SOUND(-50,2000,26)
970 CALL SOUND(-50,4000,25)
980 CALL CHAR(101,"0000000F")
990 RETURN
1000 DATA 070F0C0C0F3F3F0F,0000000F,030F3F3F3F3F3F3F,3F3F3F3F073F073F,
073F1F0F07
0301,000000000303F3F
1010 DATA FCFEFEFEE2FAE2FA,E2FEFEFCF8F0F0F0,FCFF0777BBBBBBDD,DDDDEEEEE
E0EFE,FFFF
FEFCF8F0F0F0,F0F0F0F0F0F0F0F0
1020 DATA 000003C7EDBFFFF,0000000307F0F1F3F,663C3CFFFFFFFFFF,000000C0E
0F0F8FC,00
0001030303030303,7BF3E3C181010101
1030 DATA FFFFFFFFFFFFFFFF,DECEC68686868686,FF,EF0F0F07010101,BFC04,01
0202040F08
1012,FF000000FF0000AA
1040 DATA FF000000FF0F1F9,FC0C0C1CFCF8F83,2025404A8080FF,005500AA0000F
F,20204041
8183FF,70E0E0C0C0808
1050 FOR H=1 TO 7
1060 CALL SOUND(80,RND*50+110,25,2000-RND*50,25,-3,24)
1070 CALL SOUND(80,RND*50+110,25,2000-RND*50,25,-3,24)
1080 NEXT H
1090 RETURN
```

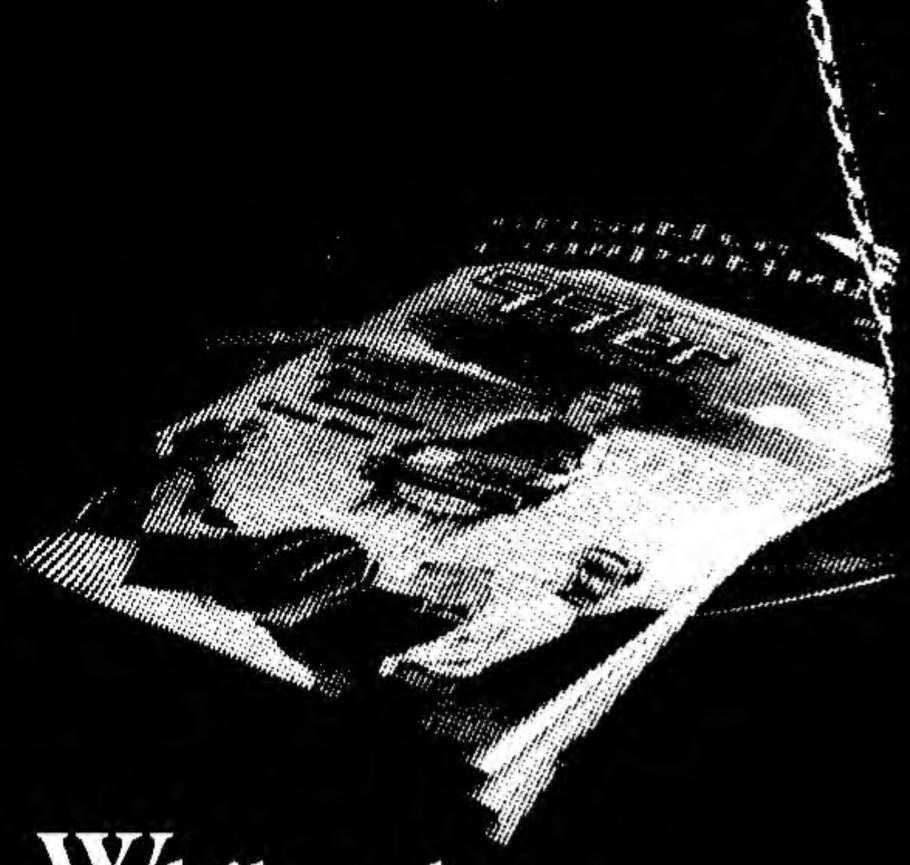


```
1100 PRINT "              ";CHR$(112):"              ";CHR$(113);CHR$(114);
CHR$(115)
1110 PRINT "               ";CHR$(116);CHR$(117);CHR$(118);CHR$(119)
1120 PRINT "             ";CHR$(120);CHR$(121);CHR$(122);CHR$(120);CHR$(12
0);CHR$(12
0)
1130 PRINT "             ";CHR$(123);CHR$(124);CHR$(125);CHR$(126)
1140 PRINT "             ";CHR$(127);CHR$(128);CHR$(129);CHR$(130)
1150 PRINT :::
1160 RETURN
1170 REM
1180 PRINT #1:"WHAT IS"
1190 IF AW1<10 THEN 1290
1200 IF AW1<>10 THEN 1230
1210 PRINT #1:"TEN"
1220 GOTO 1300
1230 IF AW1<>11 THEN 1260
1240 PRINT #1:"ELEVEN"
1250 GOTO 1300
1260 IF AW1<>12 THEN 1290
1270 PRINT #1:"TWELVE"
1280 GOTO 1300
1290 PRINT #1:AW1
1300 PRINT #1:"DIVIDED BY"
1310 IF AW2<10 THEN 1410
1320 IF AW2<>10 THEN 1350
1330 PRINT #1:"TEN"
1340 GOTO 1420
1350 IF AW2<>11 THEN 1380
1360 PRINT #1:"ELEVEN"
1370 GOTO 1420
1380 IF AW2<>12 THEN 1410
1390 PRINT #1:"TWELVE"
1400 GOTO 1420
1410 PRINT #1:AW2
1420 RETURN
1430 REM
1440 CALL SCREEN(3)
1450 FOR I=1 TO 5
1460 CALL SOUND(-100,880,0)
1470 CALL SOUND(-100,880,8)
1480 CALL SOUND(-100,880,16)
1490 CALL SOUND(-100,880,26)
1500 NEXT I
1510 CALL SCREEN(4)
1520 RETURN
1530 REM
1540 CALL SCREEN(9)
1550 FOR I=1 TO 5
1560 CALL SOUND(-100,110,0)
1570 CALL SOUND(-100,110,8)
1580 CALL SOUND(-100,110,16)
1590 CALL SOUND(-100,110,27)
1600 NEXT I
1610 CALL SCREEN(4)
1620 RETURN
```

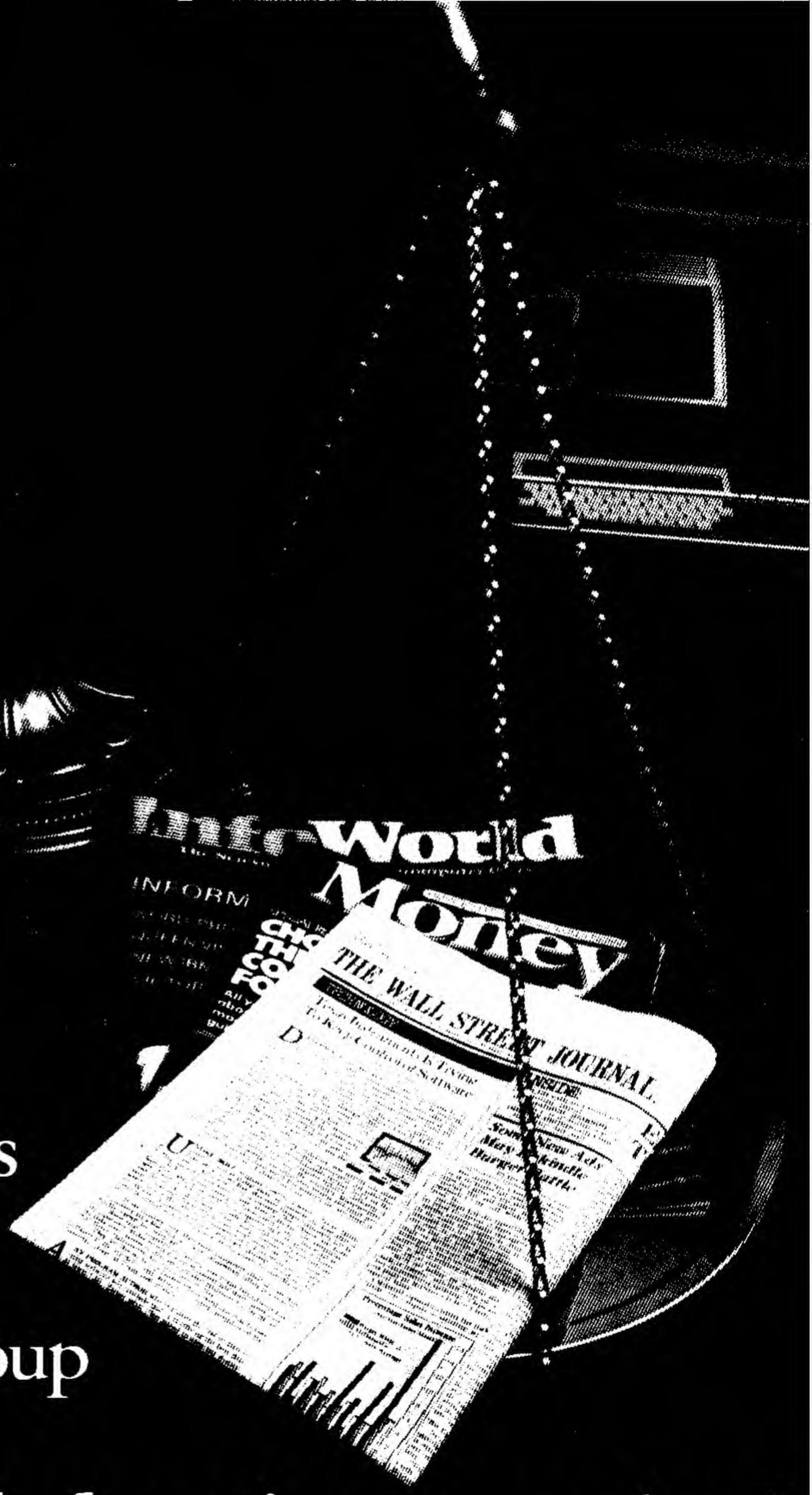


# WOMAN'S VIEW PROGRAM

```
100 CALL CLEAR
110 DIM C(4)
120 PRINT "    ** QUILT SQUARES **":
    :TAB(10);"BY REGENA"
130 PRINT : : :"USE THE ARROW KEYS
    TO MOVE": :"TO THE QUILT
    SQUARE DESIRED,": :"
    THEN PRESS <ENTER>."
140 PRINT : :"FILL IN THE SAMPLE
    SQUARES.": : :
150 CALL CHAR(91,"FF8080808080808")
160 CALL CHAR(92,"FF0101010101010")
170 CALL CHAR(93,"8080808080808080FF")
180 CALL CHAR(94,"0101010101010101FF")
190 CALL CHAR(96,"FF8080808080808")
200 CALL CHAR(97,"FF0101010101010")
210 CALL CHAR(98,"8080808080808080FF")
220 CALL CHAR(99,"0101010101010101FF")
230 FOR I=104 TO 152 STEP 8
240 CALL CHAR(I,"FFFFFFFFFFFFFFFF")
250 CALL CHAR(I+1,"0")
260 CALL CHAR(I+2,"7F3F1F0F070301")
270 CALL CHAR(I+3,"80C0E0F0F8FCFEFF")
280 CALL CHAR(I+4,"FFFEFCF8F0E0C080")
290 CALL CHAR(I+5,"0103070F1F3F7FFF")
300 NEXT I
310 CALL COLOR(10,16,2)
320 CALL COLOR(11,3,13)
330 CALL COLOR(12,5,6)
340 CALL COLOR(13,7,8)
350 CALL COLOR(14,11,14)
360 PRINT "HOW MANY COLORS (2, 3, 4)?"
370 CALL KEY(0,K,S)
380 IF (K<50)+(K>52)THEN 370
390 N=K-48
400 CALL HCHAR(23,30,K)
410 PRINT : :"1  2  3  4  5  6  7  8
    9  0": :
420 CALL HCHAR(23,3,104)
430 CALL HCHAR(23,6,105)
440 CALL HCHAR(23,9,112)
450 CALL HCHAR(23,12,113)
460 CALL HCHAR(23,15,120)
470 CALL HCHAR(23,18,121)
480 CALL HCHAR(23,21,128)
490 CALL HCHAR(23,24,129)
500 CALL HCHAR(23,27,136)
510 CALL HCHAR(23,30,137)
520 PRINT : :"FIRST COLOR?"
530 GOSUB 2060
540 C(1)=A
550 PRINT :"SECOND COLOR?"
560 GOSUB 2060
570 C(2)=A
580 IF N=2 THEN 660
590 PRINT :"THIRD COLOR?"
600 GOSUB 2060
610 C(3)=A
620 IF N=3 THEN 660
630 PRINT :"FOURTH COLOR?"
640 GOSUB 2060
650 C(4)=A
660 CALL CLEAR
670 FOR I=1 TO 4
680 PRINT TAB(7);"[\[\[\[\":TAB(7);
    "]^]^]^]^"
690 NEXT I
700 PRINT TAB(19);"1 OKAY":TAB(19);
    "2 CHANGE": : : : : :
710 ON N-1 GOTO 770,750,720
720 CALL COLOR(15,C(3),C(4))
730 CALL COLOR(14,C(2),C(4))
740 CALL COLOR(13,C(1),C(4))
750 CALL COLOR(12,C(2),C(3))
760 CALL COLOR(11,C(1),C(3))
770 CALL COLOR(10,C(1),C(2))
780 CALL HCHAR(2,21,104)
790 CALL HCHAR(2,24,105)
800 CALL HCHAR(4,21,106)
810 CALL HCHAR(4,24,107)
```



While other publications are busy quoting themselves on the TI family of personal computers . . .




. . . Look who's quoting the International 99/4 Users-Group

# Aren't you glad you're a member?

Why not tell someone else about our "quotable" group?

International 99/4 Users-Group

EDUCATION • ENTERTAINMENT • PERSONAL ENRICHMENT

```
820 CALL HCHAR(6,21,108)
830 CALL HCHAR(6,24,109)
840 RX=6
850 RY=24
860 IF N=2 THEN 1140
870 CALL HCHAR(2,27,113)
880 CALL HCHAR(4,27,114)
890 CALL HCHAR(8,21,115)
900 CALL HCHAR(10,21,116)
910 CALL HCHAR(8,27,117)
920 CALL HCHAR(6,27,122)
930 CALL HCHAR(8,24,123)
940 CALL HCHAR(10,24,124)
950 CALL HCHAR(10,27,125)
960 RX=10
970 RY=27
980 IF N=3 THEN 1140
990 CALL HCHAR(2,30,129)
1000 CALL HCHAR(4,30,130)
1010 CALL HCHAR(10,30,131)
1020 CALL HCHAR(12,21,132)
1030 CALL HCHAR(14,21,133)
1040 CALL HCHAR(6,30,138)
1050 CALL HCHAR(12,30,139)
1060 CALL HCHAR(12,24,140)
1070 CALL HCHAR(14,24,141)
1080 CALL HCHAR(8,30,146)
1090 CALL HCHAR(12,27,147)
1100 CALL HCHAR(14,27,148)
1110 CALL HCHAR(14,30,149)
1120 RX=14
1130 RY=30
1140 FOR PX=9 TO 15 STEP 2
1150 FOR PY=9 TO 15 STEP 2
1160 CALL HCHAR(PX,PY,96)
1170 CALL HCHAR(PX,PY+1,97)
1180 CALL HCHAR(PX+1,PY,98)
1190 CALL HCHAR(PX+1,PY+1,99)
1200 TX=2
1210 TY=20
1220 CALL HCHAR(TX,TY,32)
1230 TX=TX+DX
1240 TY=TY+DY
1250 CALL HCHAR(TX,TY,62)
1260 DX=0
1270 DY=0
1280 CALL KEY(0,K,S)
1290 CALL COLOR(9,16,1)
1300 CALL COLOR(9,7,1)
1310 IF K=13 THEN 1480
1320 IF K<>69 THEN 1360
1330 IF TX-2=0 THEN 1220
1340 DX=-2
1350 GOTO 1220
1360 IF K<>83 THEN 1400
1370 IF TY-3=17 THEN 1220
1380 DY=-3
1390 GOTO 1220
1400 IF K<>68 THEN 1440
1410 IF TY+3>RY THEN 1220
1420 DY=3
1430 GOTO 1220
1440 IF K<>88 THEN 1220
1450 IF TX+2=RX+2 THEN 1220
1460 DX=2
1470 GOTO 1220
1480 CALL GCHAR(TX,TY+1,G)
1490 GB=G-8*INT(G/8)
1500 ON GB+1 GOTO 1510,1510,1560,1610,1660,1710
1510 CALL HCHAR(PX,PY,G)
1520 CALL HCHAR(PX,PY+1,G)
1530 CALL HCHAR(PX+1,PY,G)
1540 CALL HCHAR(PX+1,PY+1,G)
1550 GOTO 1750
1560 CALL HCHAR(PX,PY,G)
1570 CALL HCHAR(PX,PY+1,G-2)
1580 CALL HCHAR(PX+1,PY,G-1)
1590 CALL HCHAR(PX+1,PY+1,G)
1600 GOTO 1750
1610 CALL HCHAR(PX,PY,G)
1620 CALL HCHAR(PX,PY+1,G-2)
1630 CALL HCHAR(PX+1,PY,G-3)
1640 CALL HCHAR(PX+1,PY+1,G)
1650 GOTO 1750
1660 CALL HCHAR(PX,PY,G-4)
1670 CALL HCHAR(PX,PY+1,G)
1680 CALL HCHAR(PX+1,PY,G)
1690 CALL HCHAR(PX+1,PY+1,G-3)
1700 GOTO 1750
1710 CALL HCHAR(PX,PY,G-4)
1720 CALL HCHAR(PX,PY+1,G)
1730 CALL HCHAR(PX+1,PY,G)
1740 CALL HCHAR(PX+1,PY+1,G-5)
1750 CALL HCHAR(TX,TY,32)
1760 CALL KEY(0,K,S)
1770 CALL VCHAR(17,20,62,2)
1780 CALL VCHAR(17,20,32,2)
1790 IF K=50 THEN 1160
```

```
1800 IF K<>49 THEN 1760
1810 NEXT PY
1820 NEXT PX
1830 CALL HCHAR(17,21,32,6)
1840 CALL HCHAR(18,21,32,8)
1850 FOR PX=9 TO 16
1860 FOR PY=9 TO 16
1870 CALL GCHAR(PX,PY,G)
1880 CALL HCHAR(PX-8,PY-8,G)
1890 CALL HCHAR(PX-8,PY,G)
1900 CALL HCHAR(PX,PY-8,G)
1910 CALL HCHAR(PX+8,PY-8,G)
1920 CALL HCHAR(PX+8,PY,G)
1930 NEXT PY
1940 NEXT PX
1950 FOR I=1 TO 13
1960 CALL HCHAR(19,17+I,ASC
     (SEG$("1 NEW PATTERN",
     I,1)))
1970 NEXT I
1980 FOR I=1 TO 6
1990 CALL HCHAR(20,17+I,ASC
     (SEG$("2 STOP",I,1)))
2000 NEXT I
2010 CALL KEY(0,K,S)
2020 IF K=50 THEN 2220
2030 IF K<>49 THEN 2010
2040 CALL CLEAR
2050 GOTO 310
2060 CALL KEY(0,K,S)
2070 IF (K<48)+(K>57)THEN 2060
2080 CALL HCHAR(23,20,K)
2090 A=K-48
2100 IF A<>1 THEN 2130
2110 A=16
2120 RETURN
2130 IF A<>9 THEN 2160
2140 A=11
2150 RETURN
2160 IF A<>0 THEN 2190
2170 A=14
2180 RETURN
2190 IF A<>4 THEN 2210
2200 A=13
2210 RETURN
2220 CALL CLEAR
```

*(continued from page 21)*

**TITLE: QUICK COPYer**
**AUTHOR: Quality Software**
**PURPOSE: disk copying**

Quality Software has developed a new product, QUICK-COPYer™, that will enable those of you with single-sided disks to backup the entire disk in three passes or less.

If you have a single disk drive and have made full disk backups, you can see the benefits QUICK-COPYer has to offer. Depending upon how full the disk is, QUICK-COPYer will read as much of the Source disk as memory allows, then write the entire memory to the destination disk. This is a great timesaver when compared to the Disk Manager module, which backs up a disk file by file, thereby requiring the operator to "swap" disks each time an individual file is backed up.

In addition to being much faster than the Disk Manager module when backing up disks, QUICK-COPYer also proves to be easier on your disk drive.

QUICK-COPYer has been developed for use with the following: Extended BASIC, Mini-Memory, or Editor/Assembler. Due to the reduced memory available with Extended BASIC or the Editor/Assembler, full, single-sided disks will require four passes or less.

Those with double-sided disk drives can also benefit from QUICK-COPYer. A double-sided disk may be backed up in six passes or less.

QUICK-COPYer requires the 99/4 or 99/4A console, disk drive, 32K RAM, and your choice of Extended BASIC, Mini Memory or the Editor/Assembler module. The program is $39.95 plus $2 shipping and handling. QUICK-COPYer is now available from QUALITY SOFTWARE, 1884 Columbia Road, #500, Washington, D.C. 20009, phone (202) 667-3574. VISA and MasterCard are accepted.



## INDEX TO ADVERTISERS

# CHARLIE'S PAGE

**By Charles La Fara**
President, International 99/4 Users-Group

*"These are the times that try men's souls"*

Never before in the history of the International 99/4 Users-Group has this statement been more prevalent than during the past two months. Texas Instruments' October 28th decision to exit the Home Computer market has created a total state of chaos within our organization. Employees question how much longer they will be employed, consumers scramble for hardware and software products which do not exist, distributors are confused over inventory levels and are undecided as to whether to purchase more or dump the entire TI line, contracts with software publishers are in a state of limbo.

Controversy and confusion is nothing new to the IUG's relationship with Texas Instruments. We were able to weather the GREAT EXTENDED BASIC shortage of 1980, the rebate wars of 1981-82, and the third-party software producers' lockout of 1983. This year, however, Texas Instruments has added a new twist which has everyone confused. Total unpredictability!

Always before, when TI announced a product for their Home Computer, one could assume that it eventually would be produced in significant quantities to satisfy consumer demand. Now, however, no one knows what hardware and software products are available and what hardware and software products will continue to be available to over two million owners of the 99/4A Home Computer. It's as if "the lights are on, but no one's at home."

TI's current approach to phasing out of the Home Computer buisness requires them to find new sources to produce hardware and software products for their machine. In approaching this crucial matter, TI could exercise several options. The first would be to continue to produce products as long as their current piece-part inventories last. The second option would be to sell their piece-part inventories and source codes either outright or on a royalty basis to an interested third-party manufacturer. The third and least likely option would be to scrap current inventories as these were already written off during the third quarter of 1983.

Which of these options will TI choose? No one seems to know at this time or at least no one is willing to make any definitive statements regarding this matter. The pattern we are beginning to see is that TI is likely to use all three of the options, depending upon which specific piece of hardware or software title is concerned. Let's face facts. No one in their right mind would want to take on the task of building, merchandising and marketing every Solid State module in the TI Home Computer line. Start-up costs and tooling alone would bankrupt even a large company. After all, the industrial giant, TI, was unable to turn healthy profit margins after a four-year effort. Additionally, TI has an all-consuming fear that if they were to give someone exclusive rights to hardware and software production, it would somehow continue to be a negative reflection on TI's corporate image.

What, then, can be done? What does the future hold? The pattern we are beginning to see, that of TI cancelling contracts with major software vendors such as DLM, Scott, Foresman and Co., Broderbund Software, Spinnaker and Control Data Corporation would indicate that these producers will have to make a decision as to whether or not they will attempt to market their product to the consuming public on a direct basis. Several difficulties exist here in that normal distribution channels for TI products are currently disrupted and many retailers are reluctant to sell only after-market products. Additionally, to add to this confusion, retailers such as the J.C. Penney Corporation dumping hundreds of thousands of dollars of TI product at prices well below their cost, is it any wonder that the public is so confused as to what an Extended BASIC cartridge or Mini-Memory is worth at this time?

Product shortages for the TI Home Computer have already become a fact of life in many parts of the country. Items such as the Editor/Assembler, Statistics, VideoGraphs, Securities Analysis, Tax Investment and Record Keeping, Personal Real Estate, Indoor Soccer and others have seemed to disappear. A check of 34 retail outlets by the IUG staff failed to locate more than five individual pieces of the previously-mentioned products. Yet a call to the TI—CARES Helpline indicated that many of these items can be purchased directly from Texas Instruments at suggested retail prices.

Could it be that TI is trying to recoup some of its losses by price-gouging consumers with products only they have control over? If so, this is an extremely dangerous precedent, considering thousands of consumers have placed orders in good faith with mail-order companies, such as the IUG, and expect delivery on these items all of which we have had on order from our distributors for several months now. Although we have been told by TI representatives that many of these products will be developed within the next several weeks, we have no assurances how TI will distribute them to their remaining distributor and dealer base.

Over the past several weeks the IUG has refunded thousands of dollars in prepaid orders to its members simply because we were unable to get adequate assurances from Texas Instruments that certain products will be available. Currently we are holding over ¼ of a million dollars in escrow accounts for products that have been purchased by our members that we are unable to secure, due to lack of answers from TI manufacturing facilities. The only information we have been given by Texas Instruments at this time is that the following items will not be available, due to TI's exit from the Home Computer business prior to initial production runs. These items are: Advanced Assembly Debugger, TI-Pilot, TI—FORTH, Entrapment and TI Mini-Writer. Hardware which will no longer be produced is the Pascal Development System, Video Controller, HEX-BUS Interface, MBX Expansion System and MBS Joysticks and additional 99/4A Home Computers other than for supply to warranty and repair stations.

We would like to thank our many members who have been patient with us in waiting for either products or refunds and assure all of our membership that we are doing everything in our power to provide as much product as we possibly can. Those members who are still waiting for TI products or refunds, we ask that you please be patient and try to understand that we are faced with a situation which we have little or no control over.

As President of the International 99/4 Users-Group, I would like to give my personal assurance that each member of the IUG is considered a valuable asset to our organization and we would not compromise any member for our own personal gain. The IUG is committed to provide continued service exclusively for owners of the 99/4A Home Computer in the months and years to come.

# NO WAIT!

## BACK ISSUES ARE AVAILABLE




So you don't have every issue of **Enthusiast '99** — That's no reason to end it all!! Put that thing down and pick up a pen instead — Back issues are available, but quantities are limited. Order now for speedy delivery.

$2.75 each

Include check or money order (In U.S. funds only) TO:

**Enthusiast '99**
P.O. Box 67
Bethany, OK 73008

NAME: _____

ADDRESS: _____

CITY: _____

STATE _____ ZIP _____

May, July, September, November and January issues are available.

| ISSUE(S): | QUANTITY: |
|---|---|
| _____ | _____ |
| _____ | _____ |
| _____ | _____ |
| _____ | _____ |

TOTAL ENCLOSED: _____

# The new TI Professional Computer. It makes you the one with the answers.

To help today's professional and small business executive come up with answers and alternatives, we offer the Texas Instruments Professional Computer. The one with the power, the expandability, the easiest-to-use keyboard, extremely high resolution graphics, and a broad array of software.

All the leading operating systems and programming languages are available. Plus many popular application programs from the best software suppliers to help you create spreadsheets, do word processing, construct graphics, communicate with other data bases or create your own. And for a clear "hard copy" printout of information, you can rely on the new 850 Series printers, made by TI especially for the new TI Professional Computer.

The TI Professional Computer.

**SPECIFICATIONS**

**System Unit.**
16-bit, 8088 microprocessor, 64K byte RAM, expandable to 256K bytes, 4K byte graphics display memory, 5 slot expansion bus.

**Keyboard.**
Specially designed low profile, 74-key typewriter layout, 97 keys, including 12 function keys, separate numeric keypad and cursor control clusters, tactile response, for quick positive entry. "Typewriter and numerical" feliers.

**Display Units.**
12 inch monochrome (green phosphor) or 13-inch full-color, 25 lines x 80 columns. High resolution, 720 x 300 pixels.

**Mass Storage.**
Built-in 360K byte diskette standard. Additional internal storage of 360K byte diskette, or 5 or 10 Mbyte Winchester disks optional.

**Communications Options.**
300 BPS or 300/1200 BPS internal modems. TTY, 3780, 3270 BSC start-stop line (to mainframe), 3270 BSC and SNA cluster (Fall 1983).

**Applications Software.**
Over 100 programs available from the most popular software vendors such as Microsoft, Ashton-Tate, "Visi-Pro," MS Spectr, Peachtree, BPI, Lifeboat and others.

**Printers.**
Choice of 11 850 Series for most applications.

It helps you make better decisions based on better information. Because it makes you the one with the answers.

For full information and a demonstration, visit your local computer dealer, or write: Texas Instruments Inc., Dept. DCA042ER, P.O. Box 402430, Dallas, Texas 75240. Or call toll-free: **1-800-527-3500** for the location of a dealer near you.

TEXAS INSTRUMENTS

Creating useful products and services for you.



261488E